

# Fuzzy Logic Decoupled Longitudinal Control for General Aviation Airplanes

Noel Duerksen  
*Raytheon Aircraft Company, Wichita, Kansas*

Cooperative Agreement NCA1-113

December 1996

National Aeronautics and  
Space Administration  
Langley Research Center  
Hampton, Virginia 23681-0001



## ABSTRACT

It has been hypothesized that a human pilot uses the same set of generic skills to control a wide variety of aircraft. If this is true, then it should be possible to construct an electronic controller which embodies this generic skill set such that it can successfully control different airplanes without being matched to a specific airplane.

In an attempt to create such a system, a fuzzy logic controller was devised to control throttle position and another to control elevator position. These two controllers were used to control flight path angle and airspeed for both a piston powered single engine airplane simulation and a business jet simulation. Overspeed protection and stall protection were incorporated in the form of expert systems supervisors.

It was found that by using the artificial intelligence techniques of fuzzy logic and expert systems, a generic longitudinal controller could be successfully used on two general aviation aircraft types that have very different characteristics. These controllers worked for both airplanes over their entire flight envelopes including configuration changes. The controllers for both airplanes were identical except for airplane specific limits (maximum allowable airspeed, throttle lever travel, etc.). The controllers also handled configuration changes without mode switching or knowledge of the current configuration.

This research validated the fact that the same fuzzy logic based controller can control two very different general aviation airplanes. It also developed the basic controller architecture and specific control parameters required for such a general controller.

100

## TABLE OF CONTENTS

CHAPTER	PAGE
1 INTRODUCTION.....	1
1.1 Fuzzy Logic as Applied to a Reusable Decoupled Flight Control System.....	3
1.2 The Goal of this Research.....	5
2 FUZZY LOGIC CONTROL THEORY .....	8
2.1 Fuzzy Overview.....	8
2.2 Fuzzification.....	8
2.3 Application of Rules .....	11
2.3 Output Aggrigation and Defuzzification.....	12
3 TOOLS .....	16
3.1 Simulink .....	16
3.2 Fuzzy Logic Toolbox.....	17
3.3 Beechjet Simulation.....	17
3.4 Generic Piston Single Simulation .....	19
4 CONTROLLERS .....	20
4.1 Flight Path Angle Controller .....	20
4.1.1 Controller Architecture.....	21
4.1.2 Fuzzy Controller.....	23
4.2 Speed Controller.....	28
4.2.1 Controller Architecture.....	29
4.2.2 Predictive Guess Inputs .....	31
4.2.3 Digitizer .....	32
4.2.4 Engine Dynamics Adapter.....	33
4.2.5 Turbulence Detector .....	35
4.2.6 Fuzzy Controller.....	36

4.3	Envelope Protection .....	44
5	RESULTS .....	47
5.1	Beechjet Simulation Results .....	48
5.2	Piston Single Simulation Results .....	56
6	DISCUSSION .....	65
6.1	Unexpected Discoveries.....	65
6.1.1	Difficulty Obtaining Accurate and Complete Knowledge .....	65
6.1.2	Instrument Pilots Operate Like Timesharing Digital Devices .....	66
6.1.3	How Pilots Adapt to Engine Response Characteristics .....	67
6.1.4	Controller Development Compared to Human Learning .....	68
6.2	Continued Research.....	69
	REFERENCES.....	71
	APPENDIX.....	75

## LIST OF TABLES

Table	Page
1. Simulated airplane characteristics.....	18

## LIST OF FIGURES

Figure	Page
1. Some typical membership functions. ....	10
2. Basic schematic of a fuzzy logic system using the Mamdani method.....	13
3. Basic schematic of a fuzzy logic system using the Sugeno method.....	14
4. Elevator controller architecture.....	21
5. Fuzzy input sets for error in flight path angle. ....	24
6. Fuzzy input sets for rate of change of flight path error. ....	25
7. Rules and output singletons of the fuzzy inference engine controlling the elevator.....	26
8. Overview of the fuzzy inference engine for the elevator controller.....	27
9. Resulting control surface from the fuzzy inference engine for the elevator controller. ....	28
10. Power lever angle (PLA) controller architecture .....	29
11. Predictive pla (power lever angle) change subsystem.....	31
12. Digitizing subsystem of the PLA controller .....	33
13. Engine dynamics adapter circuit.....	35
14. Turbulence detector system .....	36
15. Fuzzy input sets for the airspeed error .....	38
16. Fuzzy input sets for airspeed error rate .....	39
17. Output rules for the throttle controller fuzzy inference engine .....	40



18	Output singletons in percent throttle travel per second for the fuzzy throttle controller .....	41
19.	Overview of the fuzzy throttle controller.....	42
20.	Control surface produced by the fuzzy inference engine for the throttle controller .....	43
21.	Envelope protection in the flight path controller.....	45
22.	Envelope protection in the airspeed controller.....	46
23.	Approach and miss time history for the jet .....	49
24.	Climb to level off time history for the jet.....	51
25.	High altitude high speed and overspeed protection time history for the jet.....	53
26.	Stall protection time history for the jet.....	55
27.	Approach and miss time history for the piston airplane.....	57
28.	Climb to level off time history for the piston airplane .....	59
29.	Overspeed protection time history for the piston airplane.....	61
30.	Stall protection time history for the piston airplane .....	63
31.	Legend for the plot labels of figures 23 - 30 .....	64



## **1 INTRODUCTION**

This project was performed as part of the National Aeronautics and Space Administration (NASA) Advanced General Aviation Transportation Experiment (AGATE) program. The purpose of the AGATE program is to reduce the manufacturing, training and proficiency costs associated with general aviation airplanes.

One of the areas of funded research is the development of advanced flight controls concepts. The purpose of the flight controls research is to develop a control system that works in conjunction with advanced display systems to allow a pilot with minimal training to operate safely in instrument meteorological conditions.

There are three basic types of control systems to be examined. These are stability augmentation, attitude command and fully decoupled controls.

Stability augmentation involves altering the stability characteristics, usually by electronic means. A yaw damper is an example of this. These types of systems have had limited acceptance by the pilot community because they generally reduce maneuverability or create a feeling of "heaviness" in the controls.

An attitude command system has been shown to significantly reduce pilot workload, particularly in turbulence [1]. This is a system where the pilot commands airplane attitude. Using separate control surfaces, this system can be implemented as a combination fly-by-wire / mechanical control system where the pilot directly controls

the mechanically driven surfaces while simultaneously commanding an attitude.

The fly-by-wire surfaces are then deflected as required by a fly-by-wire system to achieve

the commanded attitude. An advantage of this type of system is that the airplane can be landed with a failure in the fly-by-wire system. This advantage carries with it the liability that it requires the pilot to be trained to fly the airplane using only conventional control techniques as well as attitude command techniques.

The decoupled control system has been shown to significantly reduce pilot training time [2-4]. With this system the pilot commands climb rate, airspeed and turn rate. This system is a fly-by-wire system that does not readily lend itself to a mechanical backup. Also, to produce an airplane that requires less training time, it is highly desirable to teach the pilot only one control scheme. These two factors require that a decoupled flight control system be made highly reliable since its operation is critical to the safety of flight.

One of the problems with any fly-by-wire system is that it takes a significant amount of time and effort to tune the gains of the control system to match the response characteristics of the airplane. Also, a control system developed and tuned for a particular model cannot be expected to work on a different model even if the two models are very similar.

### **1.1 Fuzzy Logic as Applied to a Reusable Decoupled Flight Control System**

A relatively new technique for controlling a plant through a feedback control loop is the use of fuzzy logic as developed through artificial intelligence research. A controller based on fuzzy logic is less sensitive to variations in the plant than a conventional controller [5 - 8]. This characteristic may enable a fuzzy decoupled control system developed on one airplane to be moved to another model with minimal retuning requirements. It may also eliminate the need for gain scheduling as a function of flight conditions.

Another artificial intelligence technique that fits well with fuzzy logic is an expert systems supervisor. This part of the controller can be programmed to provide control boundaries such as angle of attack and airspeed limits.

If a general flight control scheme such as the fuzzy / expert system described above can be perfected, then much of the development time and expense of matching an autopilot to a specific airplane can be eliminated [9 - 11]. With a reduction in development costs, a decoupled flight control system could be practical for general aviation airplanes. The implementation of this type of system has the potential of greatly reducing the initial training and proficiency costs of operating personal aircraft. This potential reduction in training and proficiency costs was the motivation for this research.

Previous work has shown that this type of control system can work in flight vehicles. A brief summary of some of this work follows.

Reference 12 used a fuzzy logic gain scheduler instead of the standard numerical interpolation gain scheduler in a non-linear F/A-18 simulation. It demonstrated that fuzzy logic could be successfully employed in a linear control system to provide gain scheduling.

Reference 13 used fuzzy logic in a cargo ship autopilot in an attempt to reduce wear on components and improve fuel efficiency.

An adaptive fuzzy logic controller was developed using reinforcement learning in reference 14 to improve the simulated response characteristics of the space shuttle in pitch, roll and yaw tracking. Also, the performance degradation associated with a non-adaptive fuzzy controller and conventional control systems was examined.

The authors of references 15 and 16 used fuzzy logic to control bank angle and roll rate on two different aircraft models. In addition, reference 15 used a method of partitioning the state space and applying the Lyapunov stability criterion to selected rule subsets to show asymptotically stable control.

Reference 17 explored some of the limitations of fuzzy logic as applied to flight control systems. The author of this paper used it for omnidirectional range navigation in an automatic flight system for transport aircraft. In this application the fuzzy controller

required more information than was originally envisioned. Hence the controller was not as robust as was originally expected.

Reference 18 applied fuzzy control to an unmanned helicopter. The fuzzy controller consisted of two layers, a navigational layer and an attitude layer. The pilot was able to control the machine via radio control by providing one of eight commands such as hover, turn left, fly forward, etc.

Reference 19 used fuzzy logic in combination with a conventional control structure for the outer loop of an automatic flight control system which was limited to approaches to an aircraft carrier. The research was done with a simulation of an F/A-18. Human knowledge was used as a basis for the fuzzy system which augmented the conventional system. This paper was an attempt to combine the best features of fuzzy and conventional control into a single system.

Reference 20 used a fuzzy logic / expert system to create knowledge bases and membership functions for the purpose of modeling selected parts of a helicopter flight control system. For this study the outputs were then linearized for analysis.

## **1.2 The Goal of this Research**

This work was based on the hypothesis that the control scheme described above is the means by which human pilots control aircraft. A flight instructor teaches the

student the rule set (e.g. You're a little slow, add some power.) and simultaneously identifies the fuzzy membership functions (what 'a little slow' looks like and how much is 'some' power). After gaining experience in several types of aircraft, the pilot generalizes the rule set and membership functions such that he can control an unfamiliar airplane satisfactorily the first time he flies it. (As long as the machine generally responds like the other airplanes he has flown.) The expert knowledge is conveyed to the pilot via stall warning, knowledge of airspeed limits, etc.

Assuming this hypothesis is true, it should be possible to design a generic electronic flight control system based on fuzzy logic which can satisfactorily control any airplane which meets FAR part 23 or 25 handling characteristics requirements. The key to success was then extracting from a pilot and implementing in a computer the input sets, rule sets and output sets with sufficient accuracy and completeness that the electronic controller could control any general aviation airplane satisfactorily.

The purpose of this work was to demonstrate that a controller can be devised that can satisfactorily control a wide variety of FAR part 23 and 25 airplanes. Therefore this research concentrated on controlling two airplanes that are at very different positions within general aviation - a 9 passenger plus 2 crew 16,000 pound business jet (Beechjet), and a generic 6 place 2,500 pound retractable landing gear piston powered single engine airplane (Bonanza class).



Since fuzzy logic control systems are nonlinear, the usual analysis tools associated with linear control system design could not be used. The analysis was therefore done using time histories of aircraft simulations being controlled by the controllers developed in this project.

Using a simulation of a business jet, a fuzzy logic controller was developed to provide decoupled control of the longitudinal axis. This system was designed to follow a calibrated airspeed and flight path command. It also provided limited envelope protection (maximum allowable airspeed and angle of attack). After the controller was developed on the business jet it was moved to the single engine piston airplane and its performance evaluated on that airplane.

## **2 FUZZY LOGIC CONTROL THEORY**

### **2.1 Fuzzy Overview**

Fuzzy logic control can be explained as a sequence of four steps. These are fuzzification of the input variables, application of a set of rules, aggregation of the output of the set of rules and defuzzification of the aggregate output.

### **2.2 Fuzzification**

Fuzzy logic for controls is an extension of fuzzy set theory. The idea of fuzzy sets was introduced by Lotfi A. Zadeh, a professor of Computer Science at the University of California at Berkeley in 1965 [21]. The difference between conventional set theory and fuzzy set theory is that unlike conventional theory, fuzzy set theory allows an element to be partially in a set, and the degree to which it is in that set can be defined. An example of this might be the set of appropriate initial control inputs required to move a car from the right lane to the left lane of a road while traveling at 30 miles per hour. Any steering motion to the right would take the car in the wrong direction. Therefore, all of these inputs belong to the set by degree 0 (they are definitely not in the set). Turning the steering wheel more than one half turn to the left is also inappropriate at this speed, therefore these commands also belong to the set by

degree 0. Assume that the optimum command for this task using some criteria is  $20^\circ$  left. This input would have a membership of degree 1. Assume also that acceptable performance results from inputs in the range of  $\pm 10^\circ$  from this optimum. Since these are also clearly appropriate, initial inputs from  $10^\circ$  to  $30^\circ$  left are also members by degree 1. Turning the wheel  $1^\circ$  to the left will eventually cause the car to change lanes in the correct direction, however it is not a command that is normally used for this purpose. This input may be a member by degree 0.1, meaning that it is mostly not a member, but is also somewhat a member.

Membership functions then are used to define the degree of membership of a particular element. They span the domain of possible elements and generally have a range from 0.0 to 1.0. Membership functions can be trapezoidal with sharp edges, bell shaped with continuous derivatives, symmetrical or not. The discrete set is a special case of fuzzy sets where the membership function is rectangular. Although the above examples all have a degree of membership at their extremes of 0.0 this need not be the case. Figure 1 shows some examples of typical membership functions [22-24].

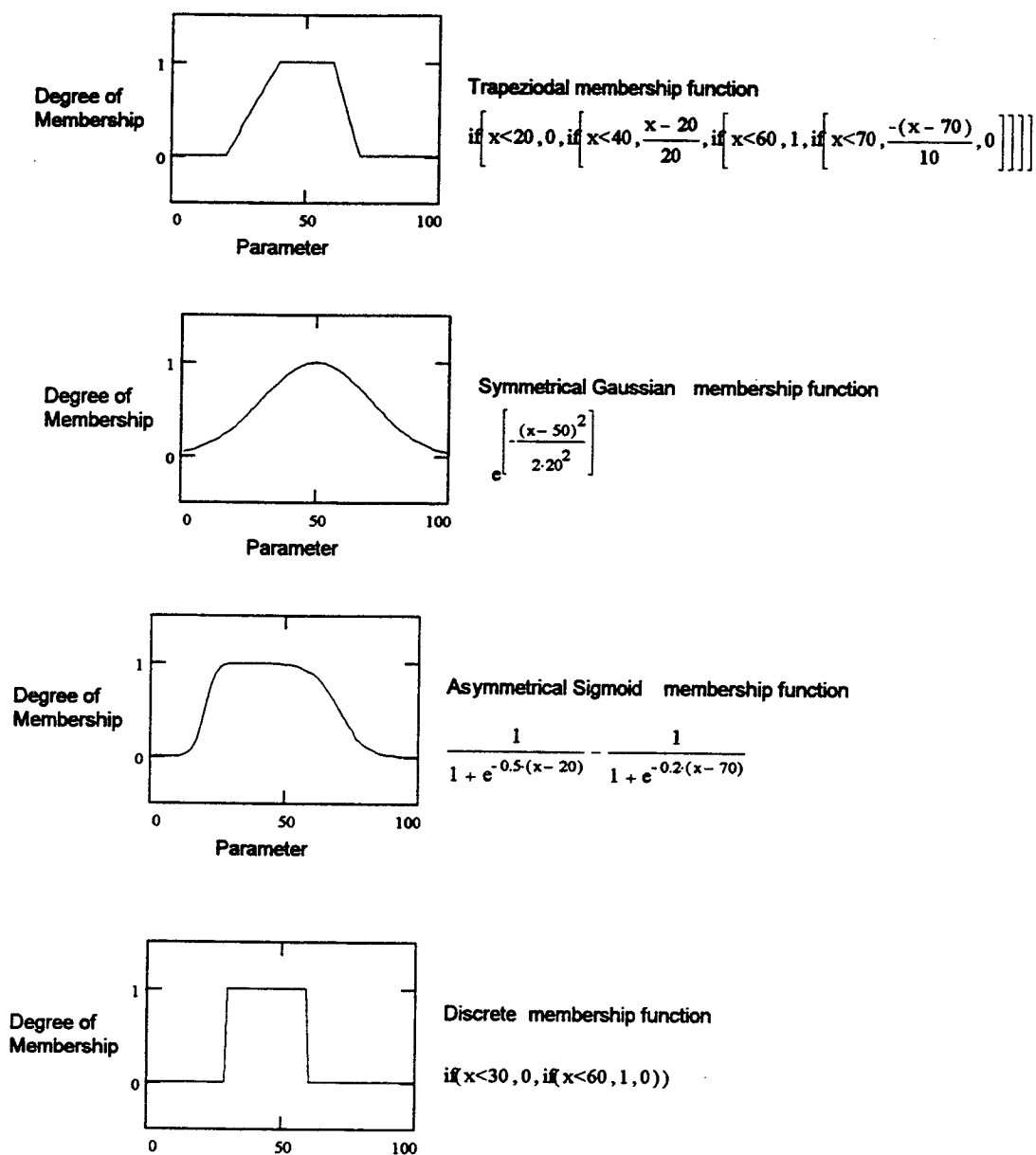


Figure 1. Some typical membership functions.

### 2.3 Application of Rules

The rule set in fuzzy logic is a linguistic set of rules as opposed to a mathematical set of rules. For example a linguistic rule that may be used in fuzzy logic might be “If the car is going too fast, then reduce the pressure on the accelerator”. The measured parameter “speed” has a degree of membership in the set “going too fast”. To the degree that the antecedent is true (the degree to which the value of “speed” has membership in the set “going too fast”) is the degree to which the consequent of the rule (“reduce the pressure on the accelerator”) is applied. Notice that the rule consequents, as well as the antecedents may be fuzzy sets.

It is postulated that humans control complex processes using this type of reasoning. It is obviously true that we pass on the skill to operate complex processes using precisely this type of linguistic structure.

Rule antecedents may have multiple inputs such as “If the car is going too fast and there is no pressure on the accelerator, then apply pressure to the brake pedal”. To handle these types of rules the equivalent of the discrete set operators  $\cup$  (or) and  $\cap$  (and) must be defined for fuzzy sets. There are many ways this can be done, and at this point in time there are no universally accepted standards. However, one of the most common is the use of the function  $\min(a,b)$  for intersection ( $a \cap b$ ) and the function  $\max(a,b)$  for union ( $a \cup b$ ) [22].

Rules may also have weights compared to each other. For instance, the rule “If there is an obstacle in the road then apply hard pressure to the brakes” may have a weight of 100 and all other rules have a weight of 1.

### **2.3 Output Aggregation and Defuzzification**

After all the rules have been evaluated, the results need to be combined or aggregated. The two most popular methods of aggregating the outputs are Mamdani’s method [25] and the newer Sugeno’s (or Takagi-Sugeno-Kang) method [26].

Using the Mamdani method, the output sets are fuzzy with membership functions defining the degree of membership for each possible discrete output. The outputs are combined by truncating each consequent membership function at the level of its corresponding antecedent degree of membership, then combining all the truncated antecedent membership functions.

Defuzzification of the aggregate output set is done by finding the centroid of the area defined by the aggregation process. The output is the value of the domain at the centroid. [22]. Figure 2 shows an example of aggregation and defuzzification using the Mamdani method.

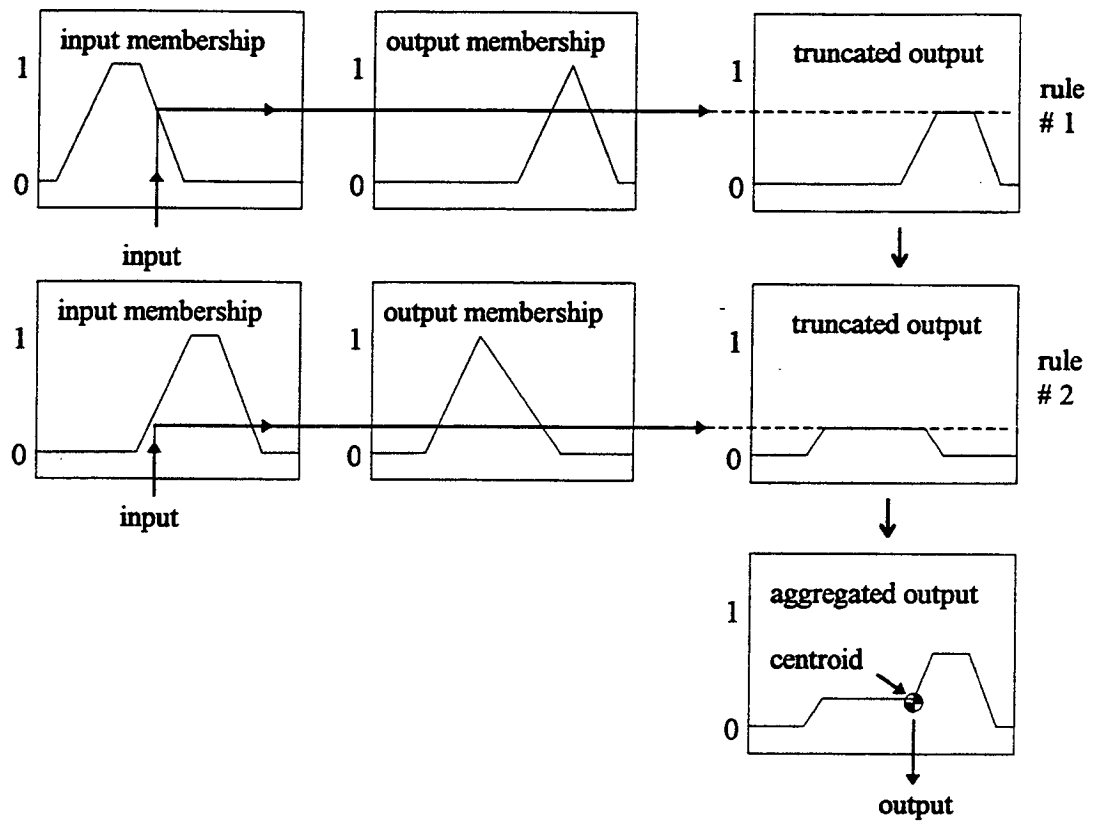
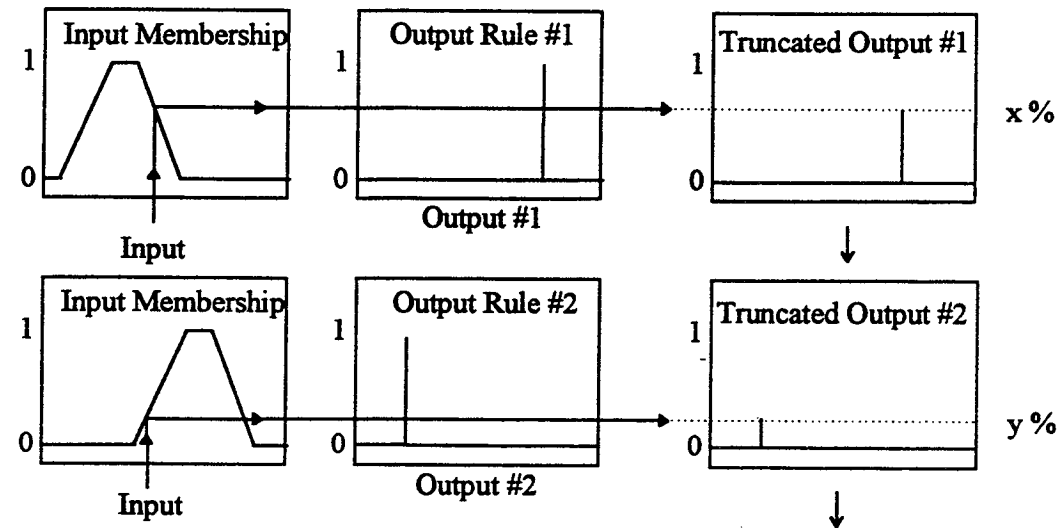


Figure 2. Basic schematic of a fuzzy logic system using the Mamdani method.

The Sugeno method is the same as the Mamdani method except that the output membership functions are not fuzzy, but are instead singleton spikes which may be movable. The output for a specific rule has the form

$$\text{rule output} = k1 * \text{rule input} + k2$$



$$[ (\text{Output Rule \#1} * x \%) + (\text{Output Rule \#2} * y \%) ] / (x \% + y \%) = \text{Final Output}$$

Figure 3. Basic schematic of a fuzzy logic system using the Sugeno method.

The aggregated output is then the weighted average of all the individual rule outputs, the weights of which are the degree of membership for the corresponding input(s) for that rule. The ability of the singleton to move via the  $k_1$  term is the equivalent of gain scheduling or mode switching. It may be noted that if the membership function is 1.0 over the entire domain, and the  $k_2$  term is zero, the result is a classic set of multi-input multi-output linear control laws. Thus, it can be said that classic control theory is a special case of the more general fuzzy logic control theory.

It is worth noting here that fuzzy set theory was not developed primarily as a control architecture. It is really a means of mapping a set of inputs to a set of outputs. As such, it can be used for modeling physical phenomena just as it can be used to



control these phenomena. Two of the more attractive attributes of fuzzy logic systems are the inherent nonlinearity of the output, and the relative ease with which imprecise human expert knowledge can be captured and incorporated into a mathematical scheme that can then be programmed into a digital computer that requires precision.

The Sugeno method was chosen for this application because it crisply defines the limits of the output control. Since the Mamdani method uses the centroid of the output sets, the control engineer must program outputs greater than the maximum allowed in order to obtain the maximum allowed. This characteristic was found to make definition of the fuzzy outputs very difficult to obtain for flight controls.

### **3. TOOLS**

Since there are no general tools for design and analysis of non-linear control systems, and since fuzzy logic systems can take on the form of almost any type of control system, it was determined that design and analysis would be performed in the time domain [27]. To do this, the system being controlled and the controllers must be simulated. This need was met with the Simulink software and simulations of two different airplanes. The development of the fuzzy sets and rules was facilitated by the use of the Fuzzy Logic Toolbox which is an add-on to Simulink.

#### **3.1 Simulink**

Simulink was used as the simulation environment for this project. Simulink is an add-on software module to the Matlab program published by The Mathworks company. It allows the development and analysis of nonlinear simulations. It is particularly well suited to this application because the simulation is developed by drawing block diagrams on the computer screen which the software then integrates to obtain time histories of the output parameters.

### **3.2 Fuzzy Logic Toolbox**

The fuzzy logic tool box from The Mathworks was used to develop the fuzzy controllers. The fuzzy logic toolbox is an add-on module to Simulink. This module simplifies the development of fuzzy sets as well as output rules. It also has several visualization tools to aid in the selection of the parameters which define the set boundaries, and shows how the rules combine to produce the output of the fuzzy system.

### **3.3 Beechjet Simulation**

The simulation used to develop the controllers was of a Beech model 400A (Beechjet). This is a 16,000 pound business jet which uses two jet engines for propulsion (table 1 summarizes the characteristics of this airplane). The simulation was a high fidelity model matched to flight test data throughout its normal operating envelope. The model included turbulence and ground effects but no ground reaction model. Simulator architecture was similar to the piston single architecture as shown in appendix A. Aerodynamic data was in the form of look-up tables. The simulation was nonlinear and could be trimmed for steady level flight at any flight condition within its normal operating envelope. It could then be "flown" via control inputs to any other flight condition in that envelope. Longitudinal control inputs to this simulation were

control column force and throttle position. It was modified to accept elevator position as an input which would be commanded by a controller. This model was used to develop the controllers because it was available at the beginning of the project while the piston single simulation was not.

	Business Jet	Single Engine Piston Airplane
Weight	13651 lb	2500 lb
Pitching Inertia	15480 slug-ft <sup>2</sup>	1300 slug-ft <sup>2</sup>
Wing Chord	6.09 ft	5.0 ft
Wing Area	241.4 ft <sup>2</sup>	160 ft <sup>2</sup>
Wing Span	42.65 ft	30 ft
Maximum Altitude	45,000 ft	15,000 ft
Maximum Airspeed	0.78 Mach / 320 KCAS	170 knots calibrated (KCAS)

Table 1. Simulated airplane characteristics

### **3.4 Generic Piston Single Simulation**

The simulation used to check for generality was loosely modeled after a Beech King Air model 200 (12,500 pound, 7 passenger, twin engine turboprop). High fidelity King Air stability derivatives were selectively modified, and then the whole model scaled to represent the size and performance of a Bonanza. The two turboprop engine models were then replaced by a single piston engine model. Table 1 summarizes the characteristics of this simulated airplane.

The Simulink block diagram for this simulation is included in appendix A. Both simulations were integrated using a Runge Kutta numerical integration scheme with fixed time steps of 0.01 seconds. This step size was chosen after running simulations of step functions driving the elevator and observing the short period response. Time steps smaller than 0.01 all had identical responses while larger step sizes showed slightly different results.

## **4 CONTROLLERS**

There were two controllers working simultaneously to control the longitudinal axis. The first controller manipulated the elevator in response to flight path error and error rate as referenced to a flight path command. The second controller manipulated the power lever angle or throttle in response to airspeed error and error rate as referenced to a calibrated airspeed command. Both controllers had a fuzzy inference engine commanding elevator or throttle rate based on their respective inputs.

Wrapped around the fuzzy inference engines was additional logic to modify the raw inputs and output, then integrate the control rate into control position and limit the control travel as necessary.

The additional logic was required to allow an envelope protection system to be incorporated, and to facilitate the adaptive control structure needed for the engine controller. Also, it allowed tailoring of the control laws to match the turbulence level.

Both controllers and associated surrounding logic were designed to imitate a method of control used by a human pilot. Note that control rate, not control position was the output of each fuzzy controller. This was selected because a pilot does not know the position of the elevator (control column). He just knows that he needs to move it some amount in a given direction.

### **4.1 Flight Path Angle Controller**

The flight path angle controller had a fairly simple architecture and a relatively simple fuzzy inference engine. A bank parameter input was included which was not used in this research because this project only dealt with the longitudinal axis. It was incorporated as a place holder for future use when the lateral axis is included.

Notice that the error as calculated by the Sum1 block has the sign reversed from the normal convention used in state feedback control. This is because this error feeds into a linguistic type of fuzzy controller where the airplane being below the desired path is “low” and for ease of constructing the fuzzy rules was chosen to be negative. This error was then fed into the digitizer circuit which was installed to model a digital computer with a 0.05 second update rate. This update rate was chosen because it follows the rule of thumb used in autopilots that a controller should have a sample rate

at least 20 times the short period frequency for smoothness, and a short period faster than one second is unlikely for these types of airplanes.

Although this project generally attempted to model a human pilot controlling an airplane, this update rate was a departure from a human pilot who has an update rate on this control of about 0.5 to 1.0 seconds with a non-constant control response between updates (not a latching output model). It was found that hold times of 0.01, 0.05 and 0.1 had virtually identical and satisfactory results. Also, with an update rate faster than 0.5 seconds a smoother ride was obtained. For this reason it was decided to depart from the human model for this parameter.

The digitized error, error rate and the bank parameter (which was always zero for this project) were fed into the fuzzy controller which in turn output an elevator rate command.

Below this section is the envelope protection circuitry which put out a zero or non-zero command. If the command was non-zero, then the controller ignored the fuzzy output and used only the output from the envelope protection block. The block labeled Switch2 switched the output based on the value of its middle input.

The clock and Switch1 block force the controller to output a rate of zero until the controllers are turned on at 2 seconds. This allows the simulation trim routine to find the correct control positions without feedback from the controllers. It also allows the simulation to operate open loop during this time before feedback control is started.

Since the output of both the fuzzy and envelope protection parts of the controller were in degrees per second and the airplane simulations required radians, a conversion was done just after Switch2. The limited integrator was required to integrate elevator rate to obtain elevator position with the limits corresponding to the elevator stops on each airplane.



#### 4.1.2 Fuzzy Controller

Aside from the bank parameter, the fuzzy inference engine had two inputs -- flight path error and error rate. The input sets for these two parameters are shown in figures 5 and 6. Figure 7 shows a list of the rules and output singletons, while figure 8 shows an overview of the whole system. Figure 9 shows the resulting three dimensional control surface from this fuzzy inference engine.

The input set boundaries were defined by recalling experiences both flying and instructing instrument flight students. The fuzzy sets for flight path error ( $\gamma$ ) were determined first. From personal experience, five sets generally work well for this type of control system, therefore this number was chosen. Since a normal approach to landing is 3 degrees it was decided that 4 degrees would definitely be extreme, but 2 degrees was not extreme. Positive and negative errors were made symmetrical since there was no information to indicate anything else would be better. The small error sets were then chosen to span the domain from the edge of the extreme set to zero. The zero set was then chosen to span the domain between the centers of the small error sets.

It is considerably more difficult to visualize flight path error rate. For this reason, only three input sets were created for this parameter. The value of 3 degrees per second was chosen as the boundary for "definitely an error" because in one second this rate would change the flight path from a normal approach to level flight. The zero membership edge for this set was then chosen as zero since zero definitely is not an error, anything else is (at least to some extent). The "zero" set boundaries were chosen to be the edges of the "error" set boundaries by applying the same logic as was used to define the "error" boundaries but in reverse.

The rule set was very straight forward. Note that there is a one to one correspondence between antecedents and consequents in this rule set. This type of rule set was chosen because it was simple and easy to analyze. This type of rule set has the

effect of combining all the rules together with their respective strengths with "or" logic. In other words, the output is a weighted average of all the rules.

The output singletons were also determined based on the author's experience and perceptions of how much control is a "slight\_increase" and how much is an "increase". The amount of control column travel associated with each output singleton was converted to elevator motion through a representative gear ratio.

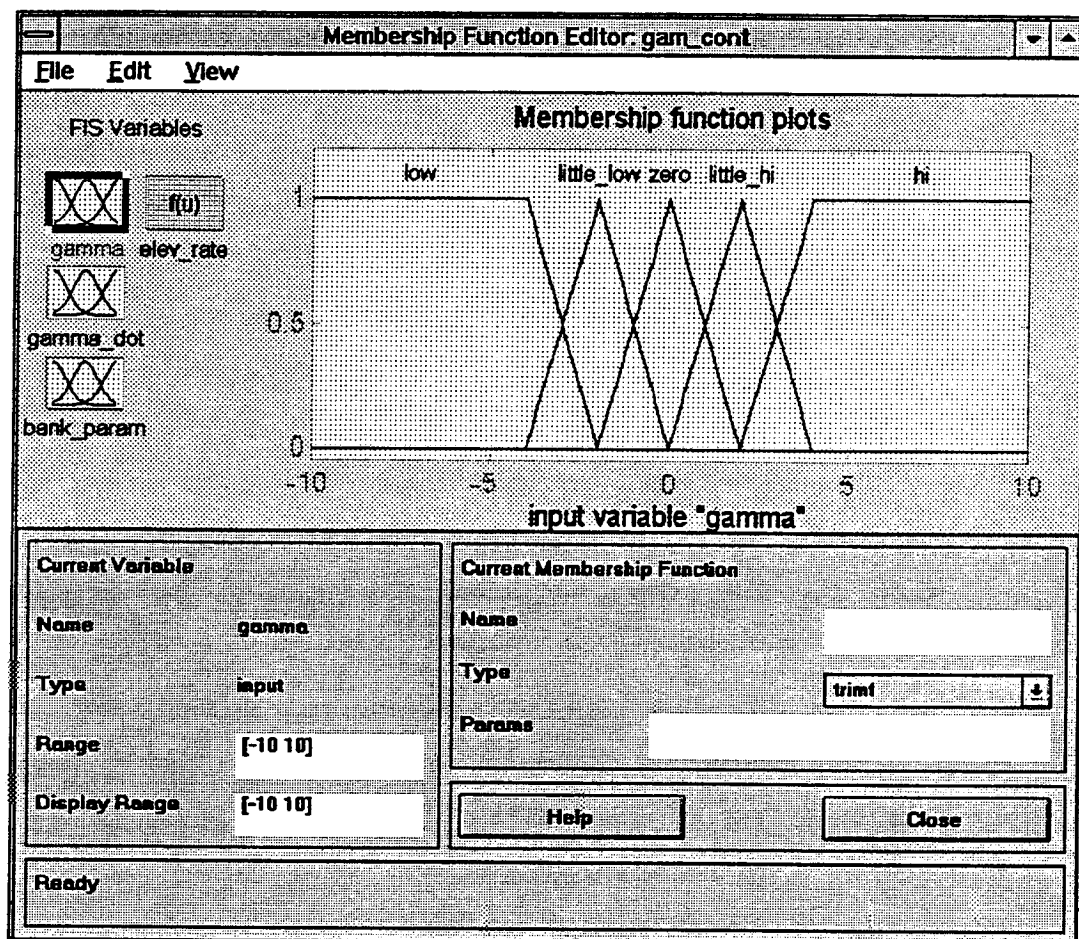


Figure 5. Fuzzy input sets for error in flight path angle.

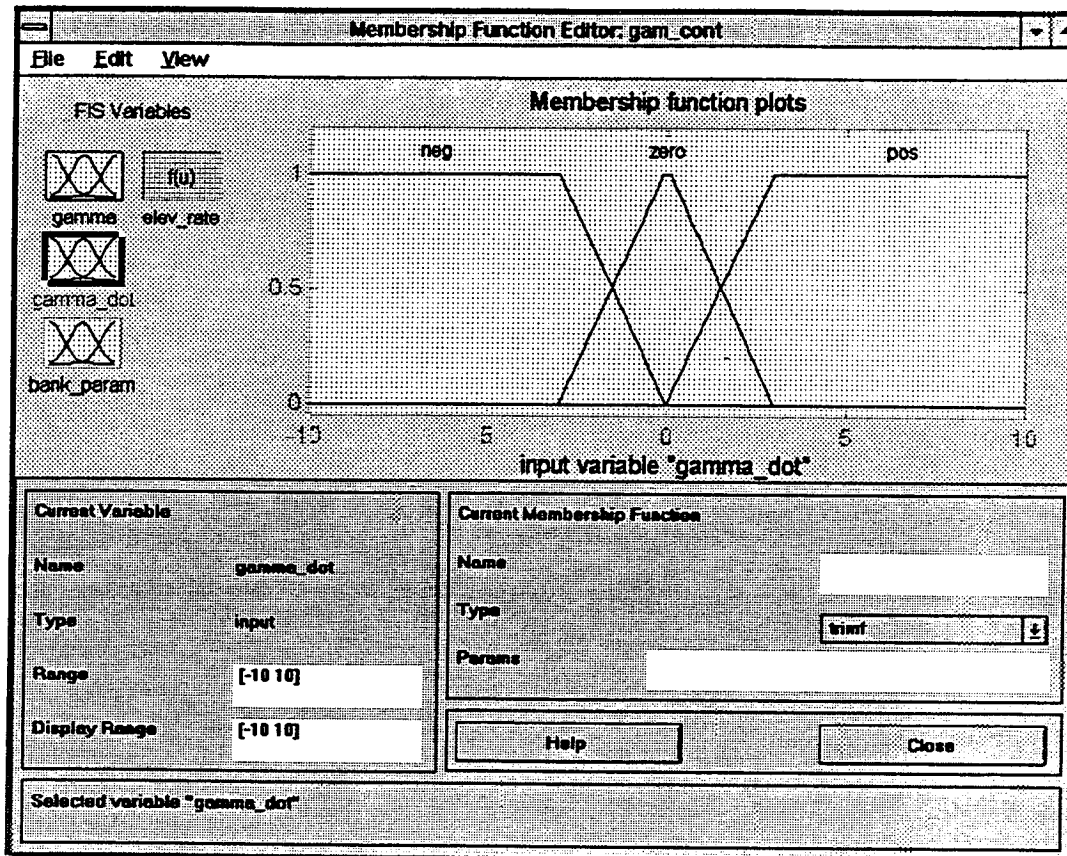
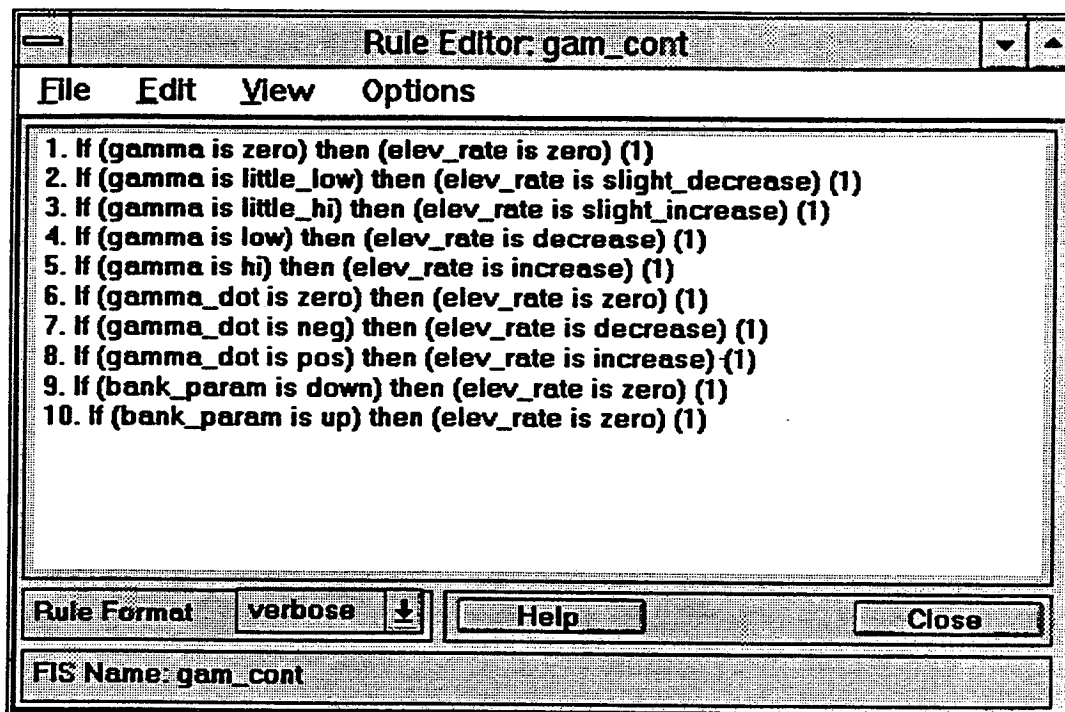


Figure 6. Fuzzy input sets for rate of change of flight path error.



Outputs are:

increase .....	10 deg / sec.
slight increase.....	3 deg / sec.
zero.....	0 deg / sec.
slight decrease.....	-3 deg / sec.
decrease .....	-10 deg / sec.

Figure 7. Rules and output singletons for the fuzzy inference engine controlling the elevator.

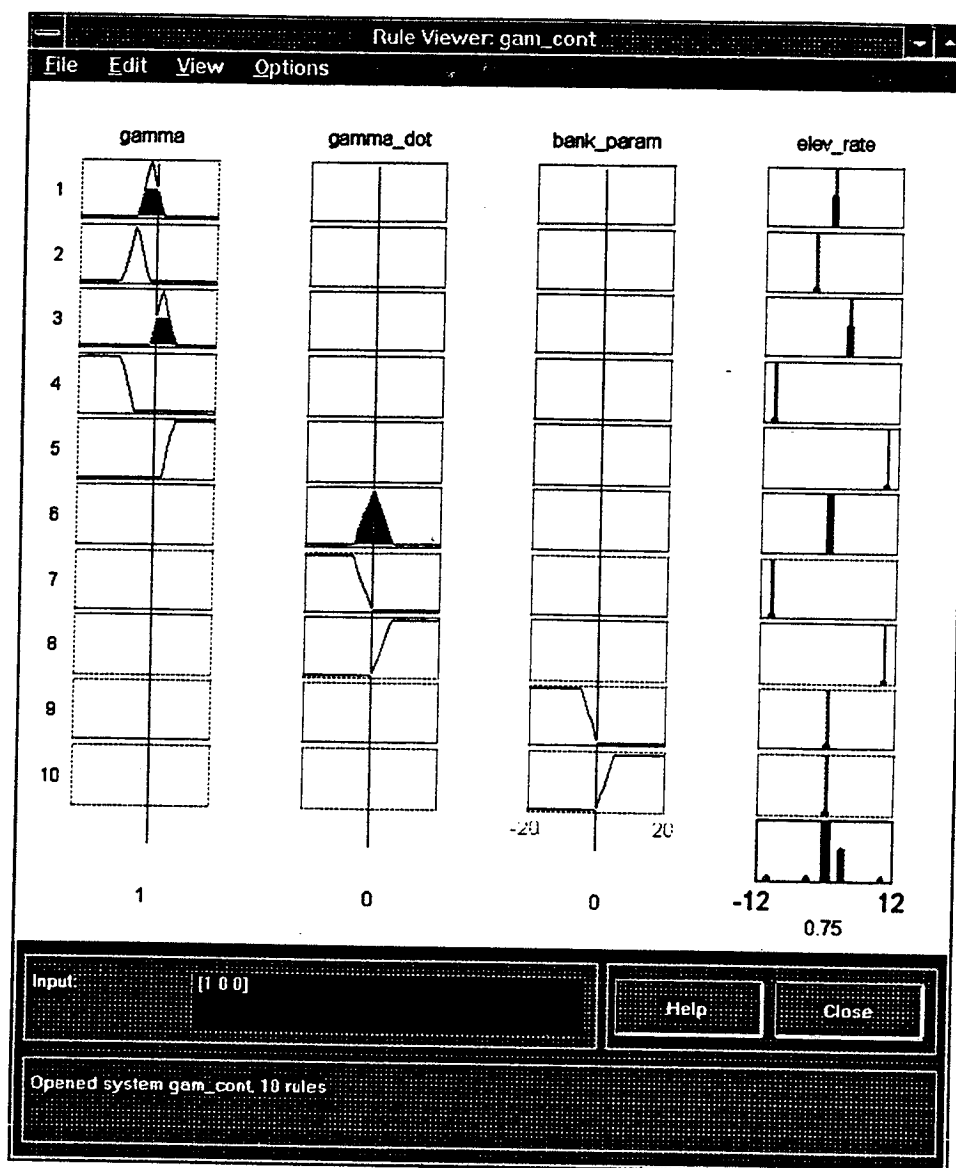


Figure 8. Overview of the fuzzy inference engine for the elevator controller. In this example there is a flight path error of +1 with 0 error rate. This caused rule #3 (as signified by the numbers in the left column) to fire at about 50% strength which in turn caused the corresponding singleton for rule 3 to fire with a weight of 50. When a weighted average of all singletons was done, the output for this condition was 0.75 deg/sec.

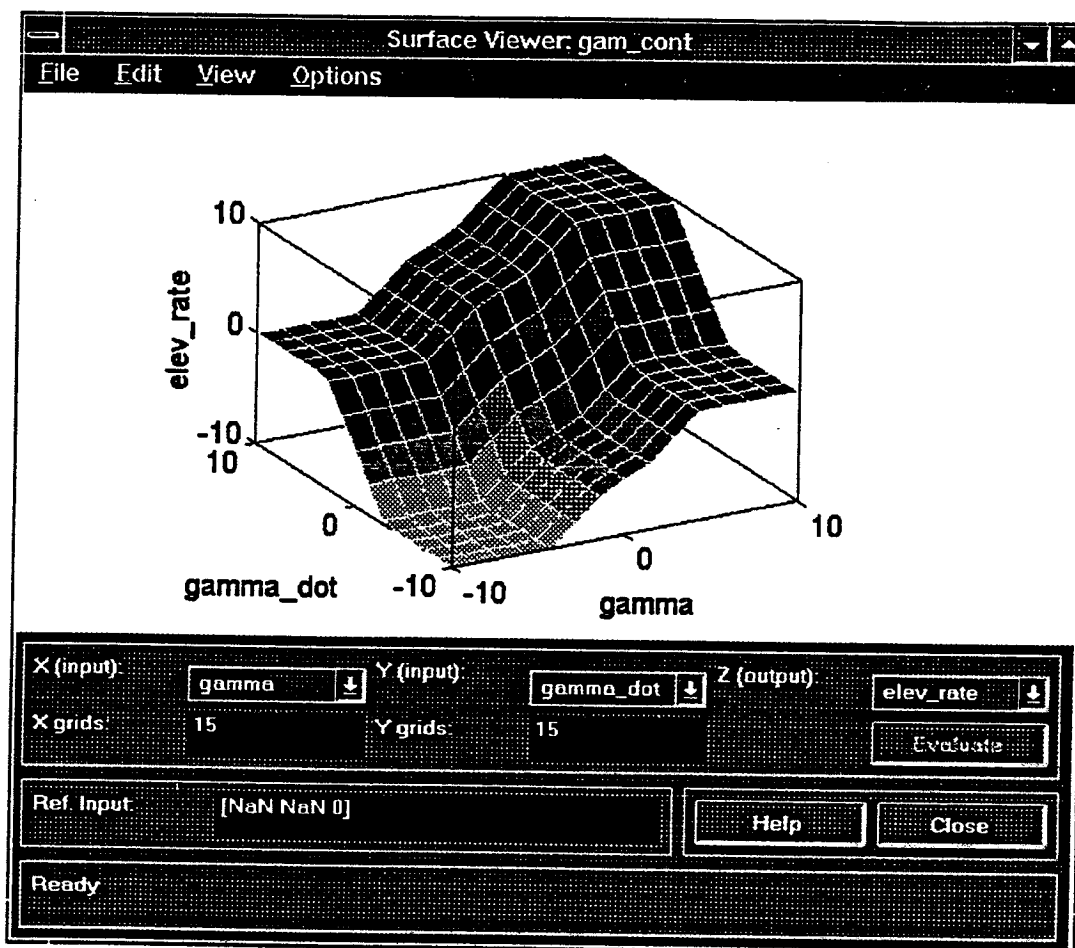


Figure 9. Resulting control surface from the fuzzy inference engine for the elevator controller.

## 4.2 Speed Controller

The speed controller was considerably more difficult to devise. It also required much more complexity than the flight path controller. A major reason for this was the fact that the two airplanes used propulsion packages with very different response characteristics. The normally aspirated piston engine with a constant speed propeller had almost instantaneous thrust response to throttle changes. The fan-jet, on the other

hand, had a considerable lag when accelerating, especially at low power or at high altitude, but a relatively fast response when decelerating.

Also, in turbulence the throttle on a jet airplane is moved the minimum amount possible because cabin pressurization transients occur with changes in engine speed. This led to the need for a turbulence detector which reduced the magnitude of the throttle response when turbulence was detected and the airplane was in otherwise unaccelerated flight.

#### 4.2.1 Controller Architecture

A block diagram of the speed controller is shown in figure 10.

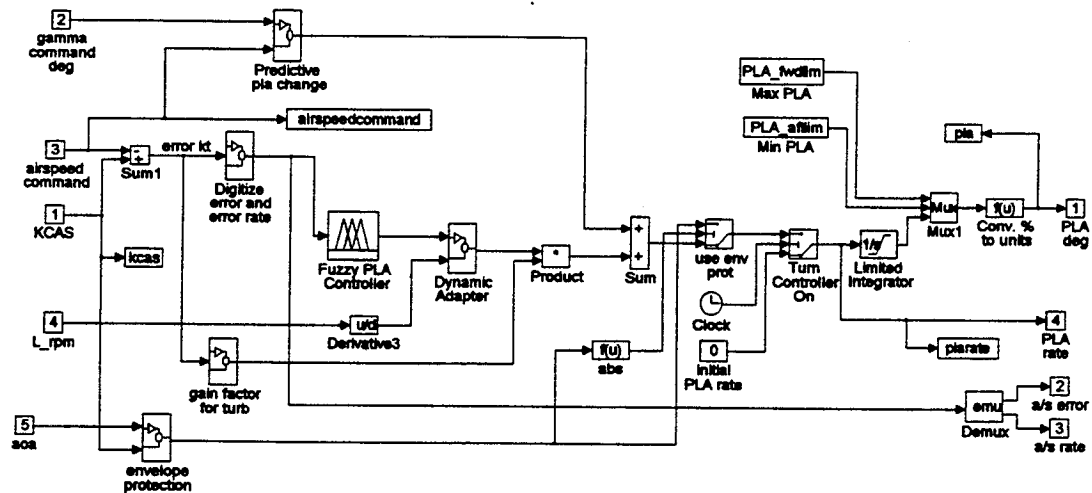


Figure 10. Power lever angle (PLA) controller architecture.

The inputs for this controller were commanded flight path angle ( $\gamma$  command deg), commanded airspeed (airspeed command), measured airspeed (KCAS), engine speed ( $L_{rpm}$ ), and angle of attack ( $\alpha$ ).

Just like the flight path controller, this controller also calculated error with the sign convention reversed from normal feedback control convention. This error was then fed into the digitizer circuit which was installed to model the way a human pilot controls the engines. The digitizer calculated error and error rate which was then fed into the fuzzy controller which in turn output a throttle lever angle rate command.

Engine acceleration and the PLA rate command are then fed into the engine dynamics adapter (Dynamic Adapter block) which selectively allowed or blocked output to the throttle based on engine dynamics.

The "gain factor for turb" block took airspeed error and used it to determine whether or not the airplane was in steady flight and the error was a result of turbulence. If it was, then this block output 0.25, if not it output 1.0. The effect of this was to reduce the throttle motion by three fourths.

The "Predictive PLA change" block used configuration changes and changes in commanded flight path and airspeed to make preemptive guesses at what the new throttle setting should be. This change was added to the modified command coming from the fuzzy controller.

Like with the elevator controller, the envelope protection circuitry for the throttle put out a zero or non-zero command. If the command was non-zero, then the controller ignored the fuzzy output and used only the output from the envelope protection block. The block labeled "use env prot" switched the output based on the value of its middle input.

The clock and Switch1 block force the controller to output a rate of zero until the controllers are turned on at 2 seconds. This allows the simulation trim routine to find



the correct control positions without feedback from the controllers. It also allows the simulation to operate open loop during this time before feedback control is started.

For generality, the control laws were in terms of percent of throttle travel. Just before exiting this system, the rate was integrated to position and percent position converted to angular position which the engine subsystem needed as an input.

#### 4.2.2 Predictive Guess Inputs

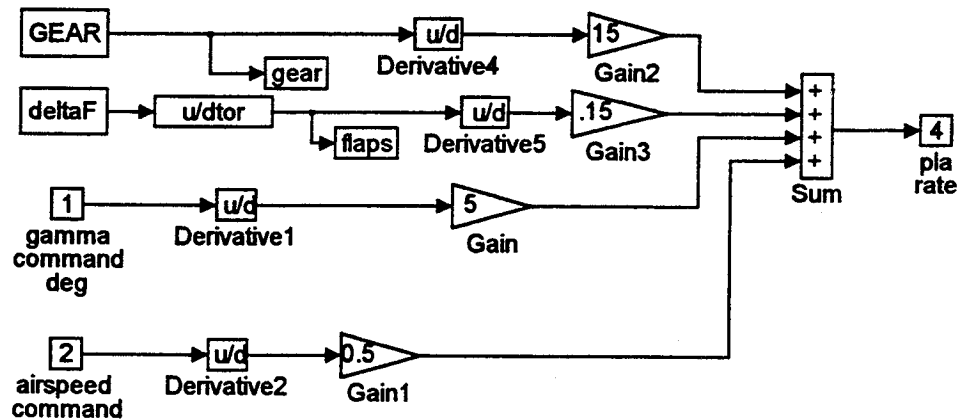


Figure 11. Predictive pla (power lever angle) change subsystem.

Figure 11 shows a block diagram of the predictive PLA change subsystem. Since engine response could be relatively slow, it was determined that for certain commanded changes, the engine controller should proactively change the throttle position instead of waiting for an error to develop (just like an experienced pilot does). The commands that triggered this response were landing gear position changes, flap position changes,

changes in commanded flight path and changes in commanded airspeed. For the sake of generality, no attempt was made to fine tune the gains associated with each of these changes. A reasonable guess of gain values was made using piloting experience. For example if an airplane is in level flight and intercepts a 3 degree glide slope, the required throttle change is more than about 10% but less than about 20%, therefore a gain was chosen such that after a commanded flight path change of 3 degrees the throttle would have moved about 15%. Also, for many airplanes if the landing gear is extended in level flight at approach speed, a flight path change of about 3 degrees results. Therefore, the gain for the gear circuit was calculated to move the throttle about 15% in response to gear extension or retraction. The same reasoning was used for commanded changes in airspeed and flap position. Note that all of these circuits use the derivative of the command. This was done to assure that the throttle change would take place smoothly over the entire time the change in command is occurring. After the initial change is made by this predictive control circuitry the normal controller function fine tunes the control based on airspeed error and error rate. Note that these control commands are added after all output inhibiting logic. This assures that they always get through undistorted and with no delay, thus mimicking a pilot's response to these command changes.

#### **4.2.3 Digitizer**

It was discovered that when flying in instrument conditions, a pilot uses a kind of timesharing technique in response to each parameter that he is required to control. Also, each of these parameters do not get equal update rates, and between updates the pilot behaves similar to a digital device which makes a step position change input then holds that position until the next update time. An update rate of once every 3 seconds

was chosen for response to an airspeed error. This update rate was chosen because this is about the rate at which an experienced instrument pilot scans the airspeed indicator.

A simple zero order sample and hold block was used to digitize the airspeed error. The difference between two successive errors divided by the sample rate was used to provide the equivalent of a digital error rate. Figure 12 shows a block diagram of this subsystem.

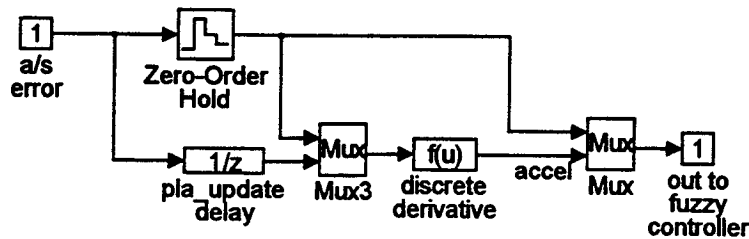


Figure 12. Digitizing subsystem of the PLA controller.

#### 4.2.4 Engine Dynamics Adapter

This subsystem was the part that made this an adaptive controller. It caused the effective rate of change of throttle position to be reduced when engine response was slow. Without this subsystem, the jet engine exhibited non-linear oscillatory behavior which was very unsatisfactory (and in some flight conditions, unstable).

There were three basic parts to this system, the throttle pulser, the stable engine detection part and the large change override circuit.

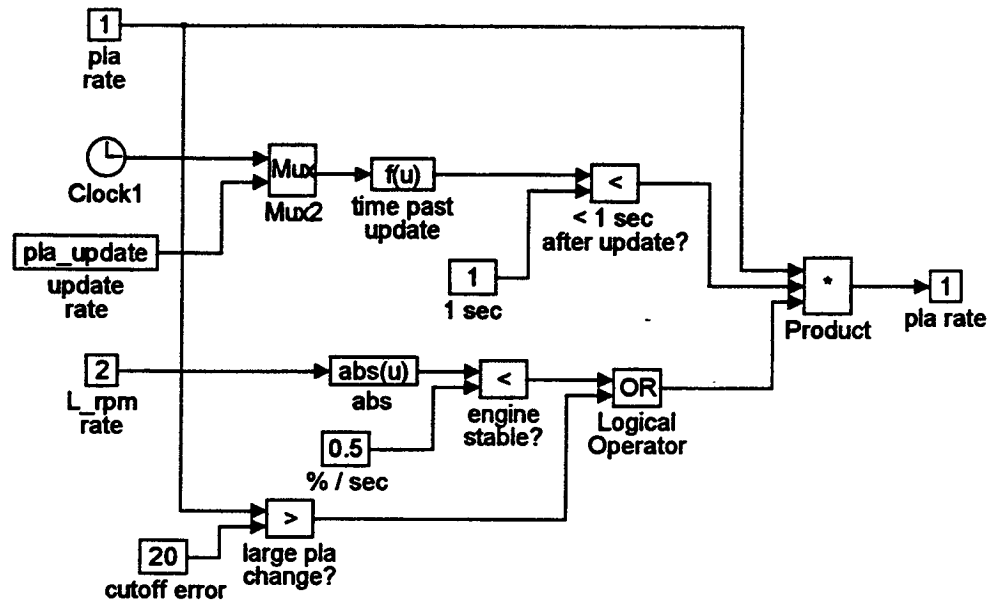
The throttle pulser caused the throttle to move in a series of ramped steps. Movement was allowed for only one second immediately after a sample of error and error rate was calculated. This was implemented as an attempt to model a human

pilot's behavior. After the pilot determines the required input, he smoothly moves the throttle and then leaves it there until the next scan cycle.

The stable engine detector did not allow a throttle movement to occur if the rate of change of engine speed was greater than a prescribed value. This prevented the oscillatory behavior of the jet engine since it effectively eliminated the phase lag caused by the slow engine dynamics. The value of 0.5% per second was chosen as the stable boundary because this is about as small a value as is practical such that turbulence will not interfere with controller operation. (Turbulence will cause some engine speed fluctuations.)

The large change override circuit was implemented to allow a large input to be made even if the engine had not stabilized. This had the effect of speeding up the response to airspeed errors without causing oscillations. This was a form of gain scheduling as a function of control magnitude where the gain schedule adapted itself to the response characteristics of the plant. The value of 20% per second was chosen as the trigger magnitude because changes smaller than that were generally the kind that caused the oscillations.

Figure 13 shows a block diagram of this subsystem.



This system allows the throttle to move only for one second after update of the input to the fuzzy controller. It also allows an input to pass if the engine is steady or requested pla change is large,

Figure 13 Engine dynamics adapter circuit.

#### 4.2.5 Turbulence Detector

Since constantly moving the throttle of a jet in response to turbulence causes cabin pressurization transients, it is considered bad piloting technique. For this reason a turbulence detector was added which reduced the magnitude of the throttle change by a factor of 4 when it detected that the airplane had reached steady state but was experiencing airspeed errors due to flying in turbulence. It was determined that the airplane had reached steady state if the airspeed error was less than 5 knots and the acceleration based on the last 20 seconds was less than 0.5 knots per second.

Figure 14 shows a block diagram of this subsystem.

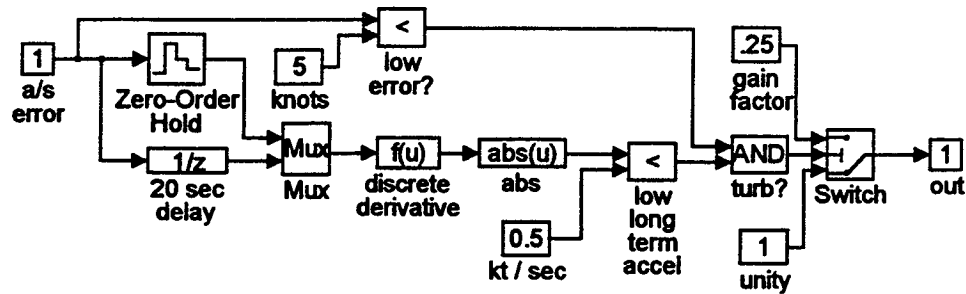


Figure 14. Turbulence detector subsystem.

#### 4.2.6 Fuzzy Controller

The fuzzy inference engine had two inputs -- airspeed error and error rate. The input sets for these two parameters are shown in figures 15 and 16. Figure 17 shows a list of the rules and figure 18 defines the output singletons (note that two of the output singletons are moving). Figure 19 shows an overview of the whole system and figure 18 shows the resulting three dimensional control surface from this fuzzy inference engine.

The set boundaries and singleton values were obtained from personal experience flying and instructing in airplanes represented by both simulations. The same techniques and thought processes were used in determining the set boundaries for the speed controller as were used for the flight path controller. The major differences were the fact that the error rate input had 5 sets for this controller while the flight path controller had 3, and the rule set was of the type where each rule had two antecedents connected by an "and" logical operator.

While flight path error rate was difficult to judge in the cockpit, airspeed error rate was not. This is because many modern business jets with electronic flight displays have an "airspeed trend vector". This trend vector usually takes the form of a magenta line on the airspeed indicator with one end at the airspeed pointer and the other end at

the airspeed that the airplane will be flying 10 seconds from now given the current acceleration. Since airspeed error rate could be estimated much more readily, 5 fuzzy input sets were developed in an attempt to provide this controller as much fidelity to a human pilot as possible.

The exclusive use of "and" rules in this rule set allowed the use of the equivalent of a 2 dimensional look-up table to define the output singletons. This form was chosen based on previous experience developing a fuzzy logic controller to regulate the fan speed of a jet engine (which worked well). The values of the singletons were chosen as a result of much "armchair flying" where an airspeed error and error rate were imagined and the corresponding throttle motion recalled.

Note that the slight error with zero error rate singletons are a function of airspeed error (they are moving singletons). This type of output was implemented because with small airspeed error and no acceleration, pilots tend to make small corrections that are proportional to error.

Unlike the flight path controller where the first set of values produced satisfactory results, this controller required some tweaking to provide satisfactory results. These adjustments were made through trial and error. The larger values seemed to be fairly robust while most of the adjustments were made to the smaller values.

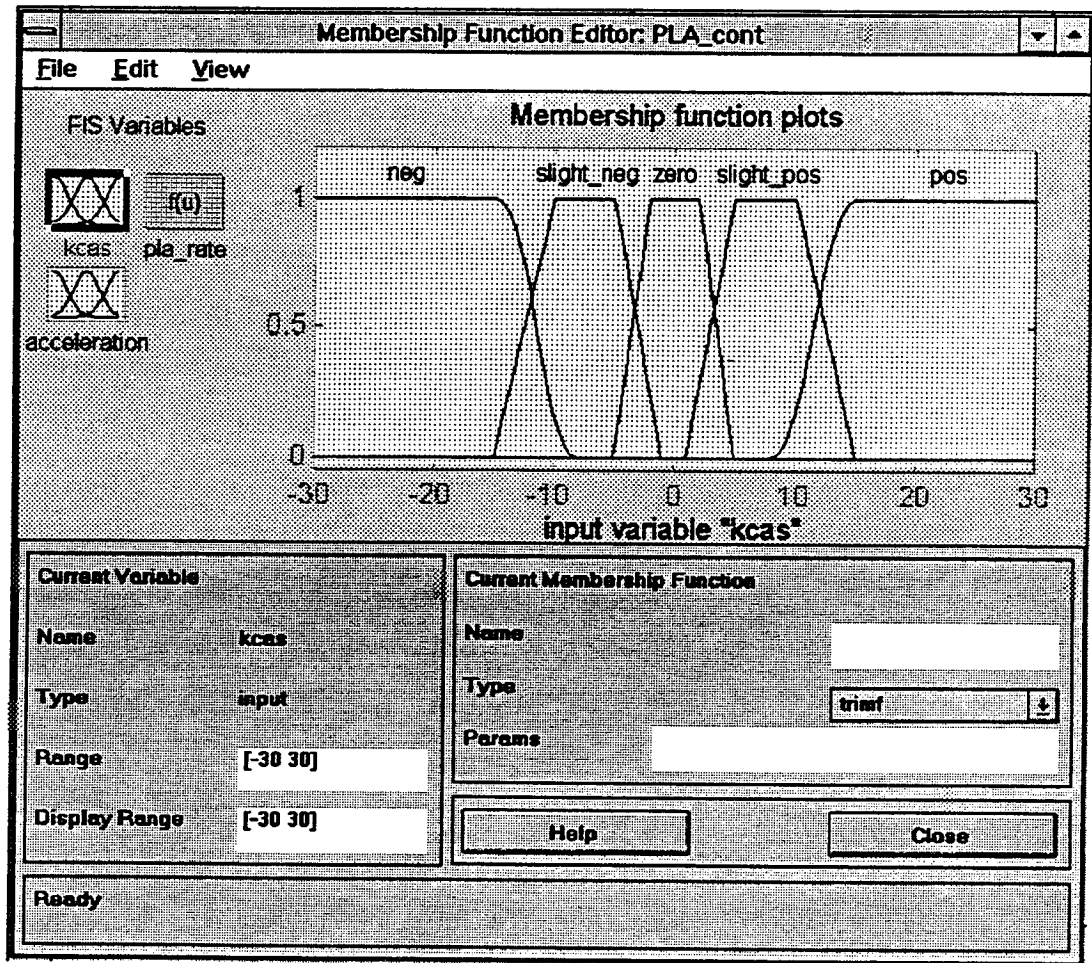


Figure 15. Fuzzy input sets for airspeed error.



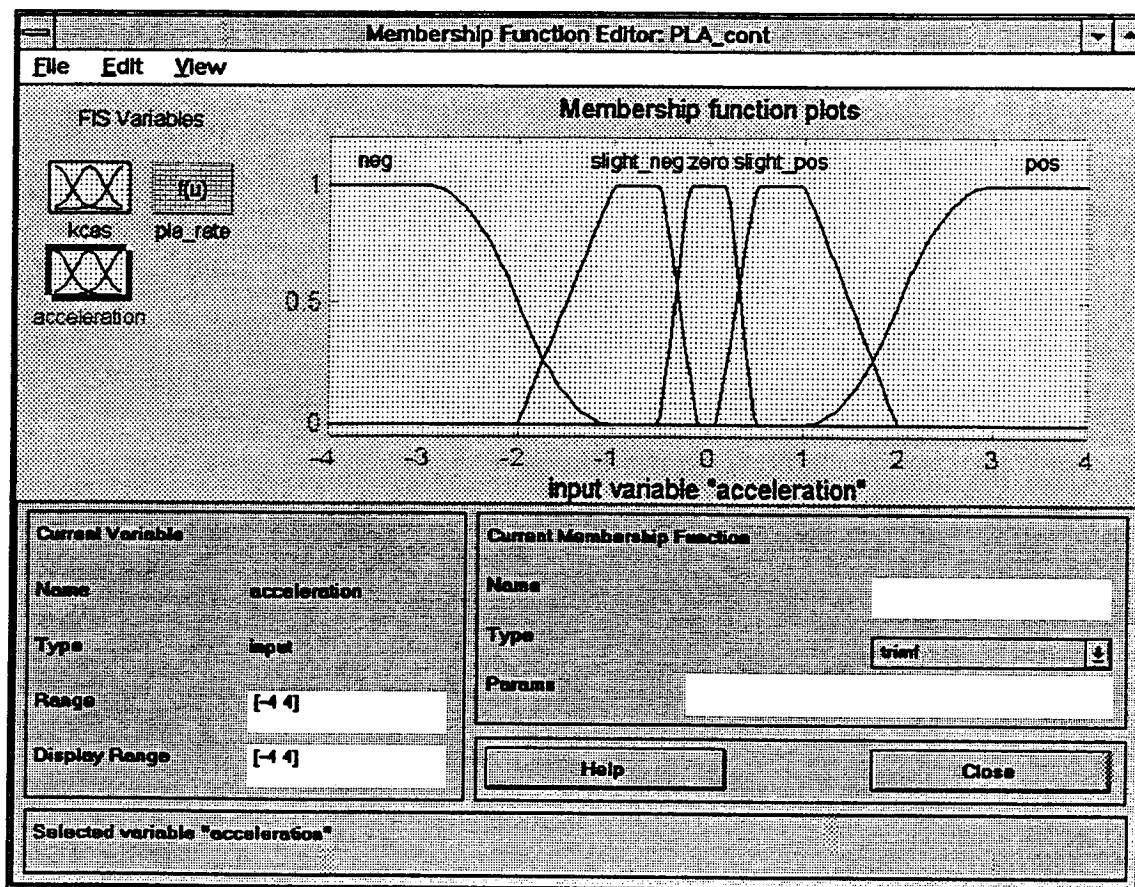


Figure 16. Fuzzy input sets for airspeed error rate.

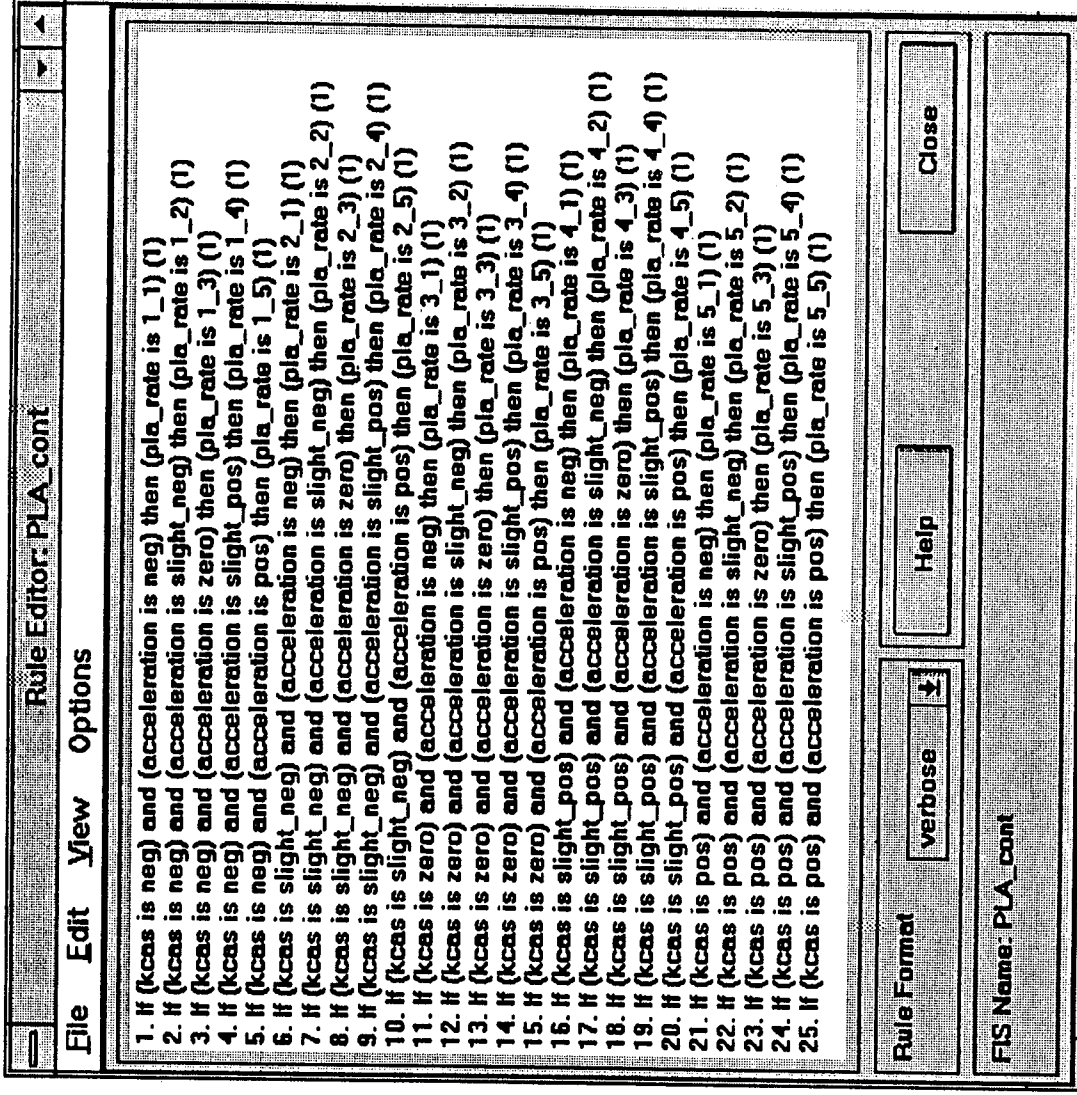


Figure 17. Output rules for the throttle controller fuzzy inference engine.

		ERROR RATE				
		neg	slight_neg	zero	slight_pos	pos
ERROR	neg	50	20	10	0	0
	slight_neg	30	10	-0.2*error	0	-6
	zero	10	4	0	-4	-10
	slight_pos	6	0	-0.2*error	-10	-30
	pos	0	0	-10	-20	-50

Figure 18. Output singletons in percent of throttle travel per second for the fuzzy throttle controller. Note that these are referred to in figure 16 by their row and column numbers. For example, rule 1\_1 is 50 and rule 3\_4 is -4.

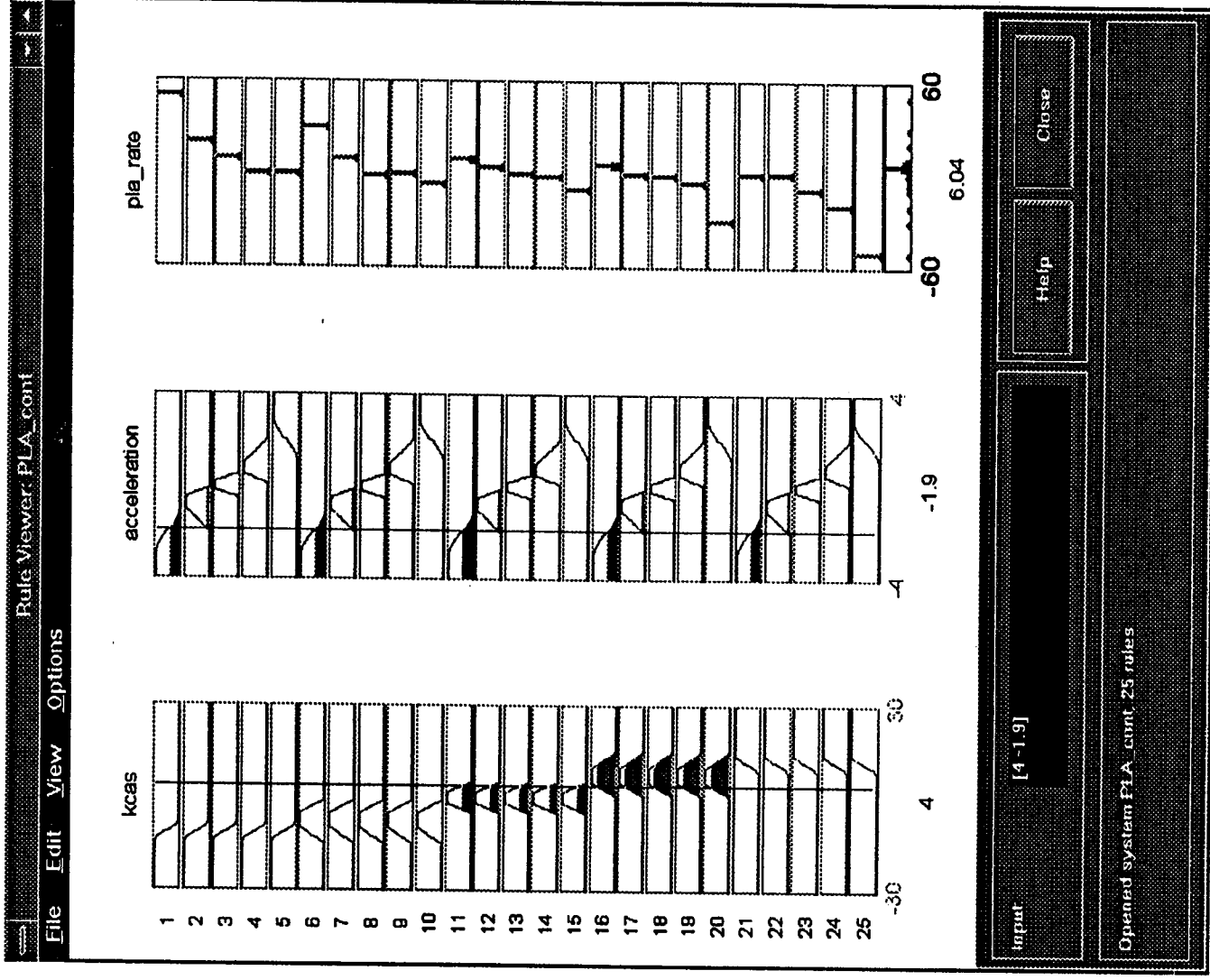


Figure 19. Overview of the fuzzy throttle controller.

See figure 8 for instructions on how to interpret.

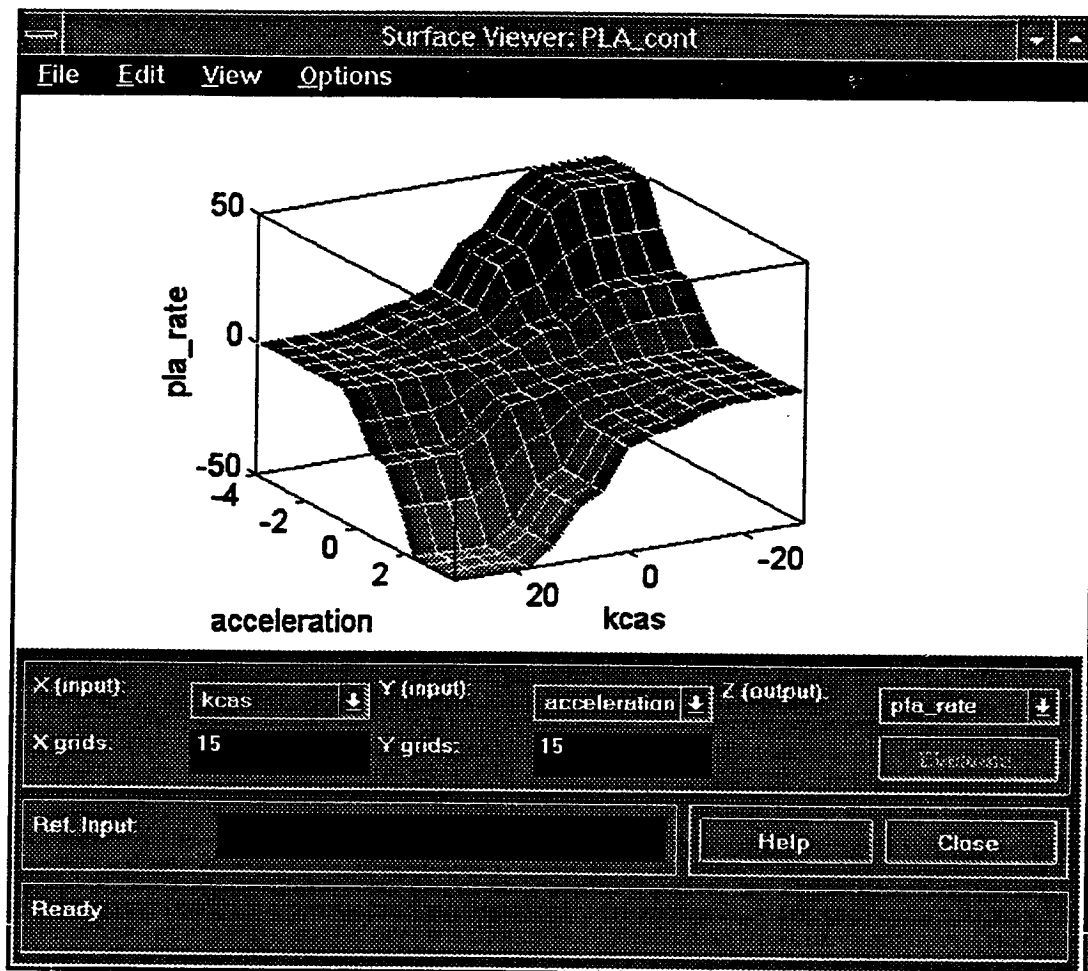


Figure 20. control surface produced by the fuzzy inference engine for the throttle controller.

### **4.3 Envelope Protection**

For the purpose of this study, only two forms of envelope protection were implemented. These were stall protection and overspeed protection. The elevator was involved in both of these. When an angle of attack above the limit was sensed, the elevator controller ignored all other inputs and moved the elevator trailing edge down at a rate of 5 degrees per second until the angle of attack was below the limit. For overspeed protection, the elevator controller sent a trailing edge up command of 5 degrees per second when the airspeed was above the limit and the power lever angle (PLA) was near the idle stop. This protected against the case where the commanded flight path angle was steep enough that even at idle the airplane would accelerate past the maximum allowed airspeed. Figure 21 shows a block diagram of the protection circuit in the flight path angle controller.

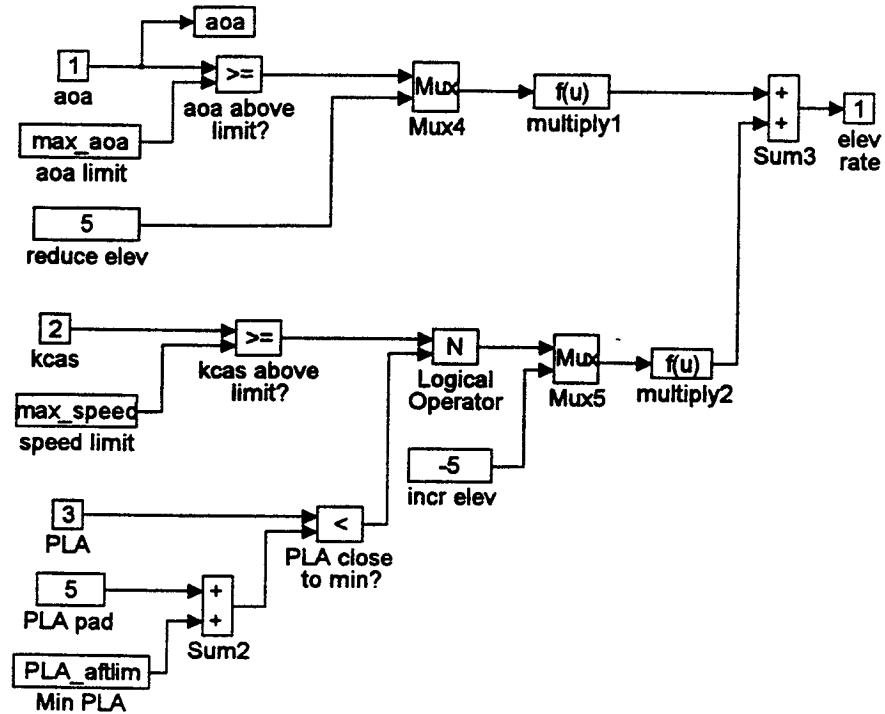


Figure 21. Envelope protection in the flight path angle controller.

The throttle was also involved with both stall and overspeed protection. When an angle of attack above the limit was sensed, the PLA controller ignored all other inputs and moved the throttle forward at a rate of 50% of travel per second until the angle of attack was below the limit. For overspeed protection, the PLA controller moved the throttle aft at a rate of 50% per second while the airspeed was above the limit. Figure 22 shows a block diagram of this circuit.

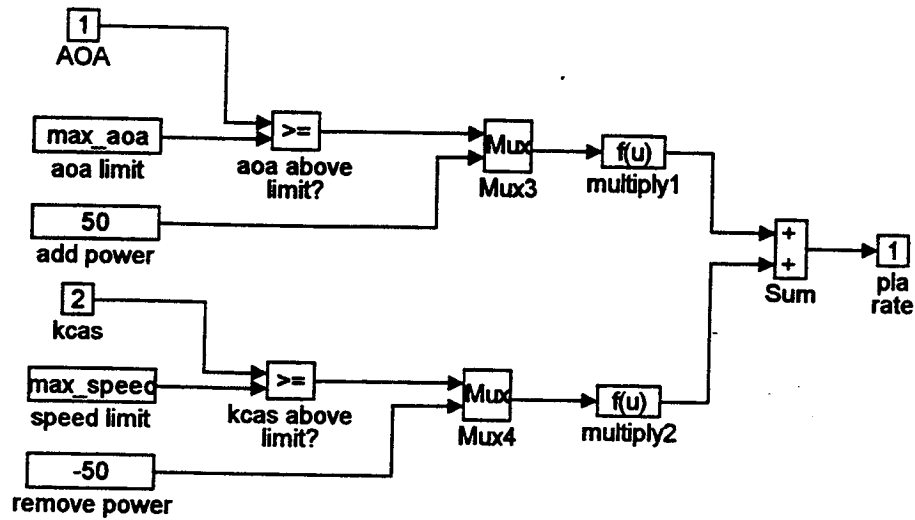


Figure 22. Envelope protection in the airspeed controller.

The elevator controller and the throttle controller were designed to work together so that the system as a whole behaves like a human pilot does during a stall recovery or inadvertent overspeed. The values of the elevator and throttle movements were chosen to reflect good piloting practice with the assumption that the angle of attack limit is set at the equivalent of stall warning, not actual stall angle of attack.



## 5 RESULTS

Simulations were run using both airplanes for the following flight maneuvers.

- 1) Level flight with gear and flaps retracted through final landing approach including extension of the gear and flaps followed by a climbing missed approach in light turbulence
- 2) Transition from level flight to a maximum power climb and back to level flight
- 3) High speed dive to exercise the overspeed protection circuit
- 4) Level flight speed reduction to exercise the stall protection circuit

In all cases the controllers worked acceptably for both airplanes when operating within the normal flight envelope. The jet airplane however, displayed more sluggish response, and overshoot characteristics than the piston powered airplane. This is interesting because these are some of the same characteristics that pilots notice most when transitioning from smaller to larger airplanes. Also, the controllers generally handled the piston airplane better than the jet airplane. This was a surprise since the jet simulation was used to develop the controllers which were then moved to the piston simulation after the design and parameters were "frozen". This fact strongly suggests that a general knowledge of how to fly an airplane was successfully implemented in the control schemes.

For many of the control parameters, the first guess worked well enough that no change was required. This leads to the suspicion that the range of acceptable values for this type of control system is fairly large. In fact, individual human pilots fly with different control parameters (styles) further validating the idea that these parameters are rather insensitive to perturbations as long as they are generally correct.

The results show that by using a fuzzy logic based control system, a wide range of general aviation airplanes can successfully use a common controller.

### 5.1 Beechjet Simulation Results

Figure 23 shows the time history of the jet simulation transitioning from initial approach to final approach and then executing a missed approach. The simulation was started with the airplane trimmed for level flight at 150 knots. At 2 seconds the controllers were turned on. An airspeed reduction to 120 knots was commanded while maintaining level flight. At about 35 seconds the landing gear was lowered, and at 40 seconds the flaps were extended. At 60 seconds a flight path command of 3 degrees down was given. At 150 seconds a missed approach was initiated by commanding a flight path of 3 degrees up. This case demonstrated the characteristics of the controllers coupled with the business jet at low speed with configuration changes.

Note that the characteristics of the speed controller caused the throttle to move in a series of pulses as is shown in by the throttle lever rate trace (PLA rate).

Also of note is the acceleration at the missed approach point. This was caused by the stall protection circuit. The airspeed was relatively slow and the change in flight path angle large enough in a short time that the maximum allowed angle of attack was reached. This caused a large and rapid throttle movement which took some time to wash out because the engine dynamics adapter waited for the engine to respond to the throttle change before allowing the next change to pass.

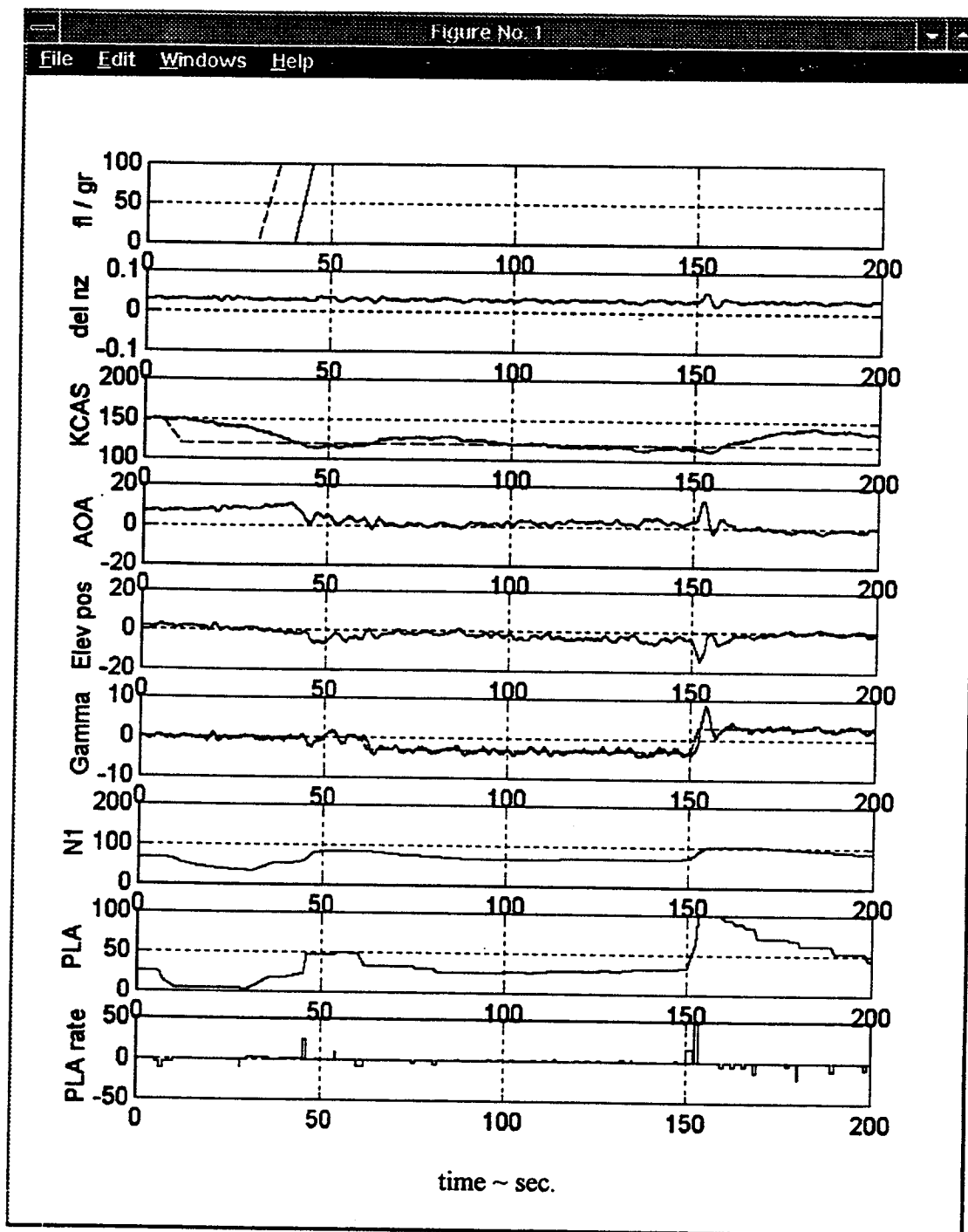


Figure 23. Approach and miss time history for the jet.

See figure 31 (page 64) for legend.

Figure 24 shows a time history of the business jet transitioning from level flight to climbing at maximum thrust and then leveling off at 35,000 feet. The airplane was initially trimmed for level flight at 35,000 feet and 250 knots (Mach 0.74). The controllers were turned on at 2 seconds. At the same time a command to climb at a 3 degree flight path angle and maintain 250 knots was given. The flight path controller followed this command and the throttle increased to maximum in an attempt to hold the commanded speed. At 100 seconds the command to level was given. The flight path controller followed this with a slight overshoot. The throttle controller reduced the throttle in response to the change in flight path command, and then moved the throttle back to maximum to accelerate the airplane back to the commanded speed. This simulation was not allowed to continue until the speed stabilized because acceleration was so slow that it was impractical to run it that long.

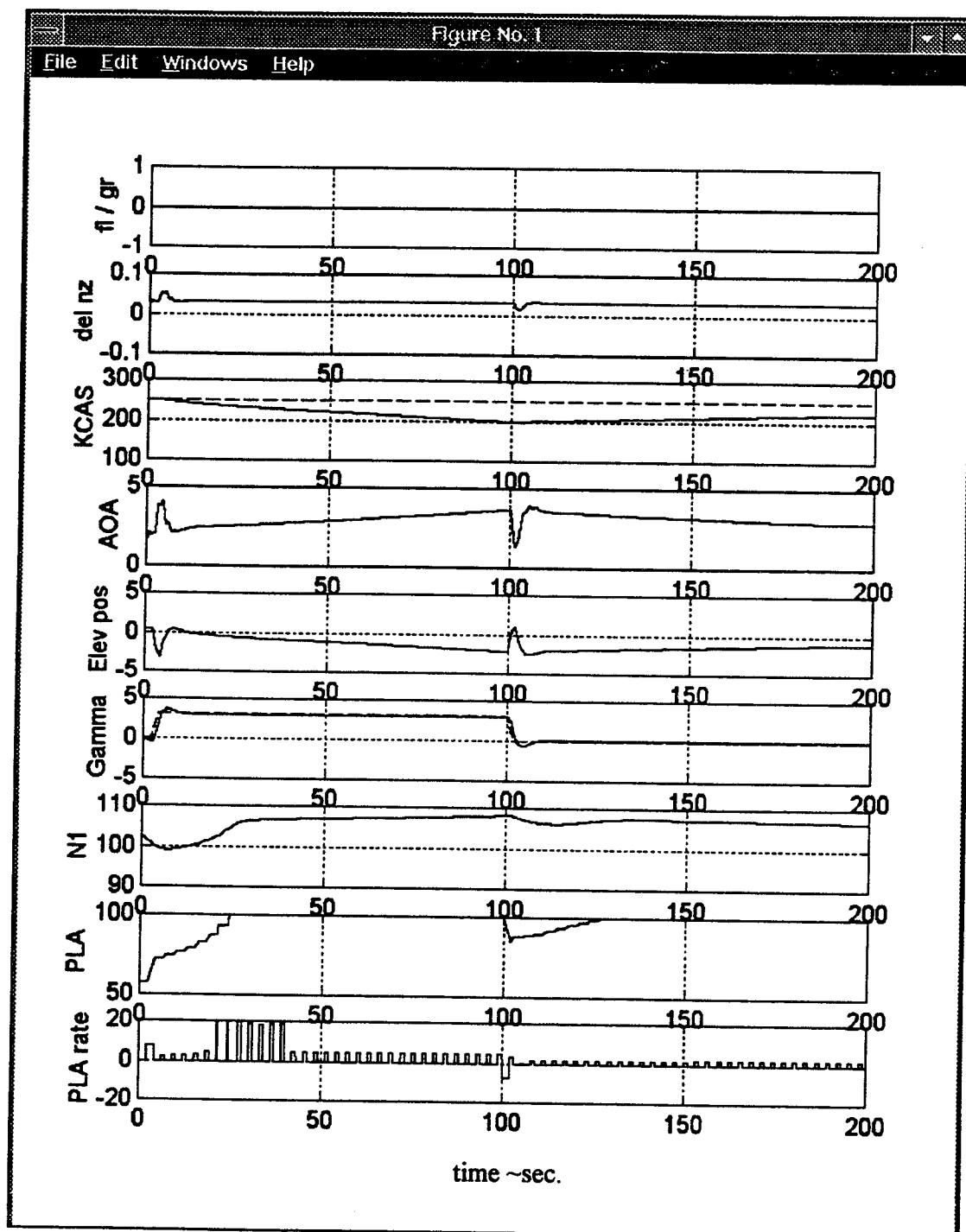


Figure 24. Climb to level off time history for the jet.

See figure 31 (page 64) for legend.

Figure 25 shows a time history of the business jet exercising the overspeed protection feature at cruise altitude. The airplane was trimmed for level flight at 35,000 feet and 250 knots (0.74 Mach). After the controllers were turned on at 2 seconds a flight path of 5 degrees down was commanded to induce an overspeed even at idle. For this demonstration the maximum speed limit was set at 260 knots (0.77 Mach). The overspeed limit was first crossed at about 27 seconds. At this time the throttle was moved rapidly to idle and the elevator moved to lessen the decent rate until the speed was again less than the limit. Since an overspeed condition did not exist from 30 to 75 seconds the elevator controller tracked the 5 degree commanded flight path. At 75 seconds the speed limit was again exceeded and the recovery process repeated.

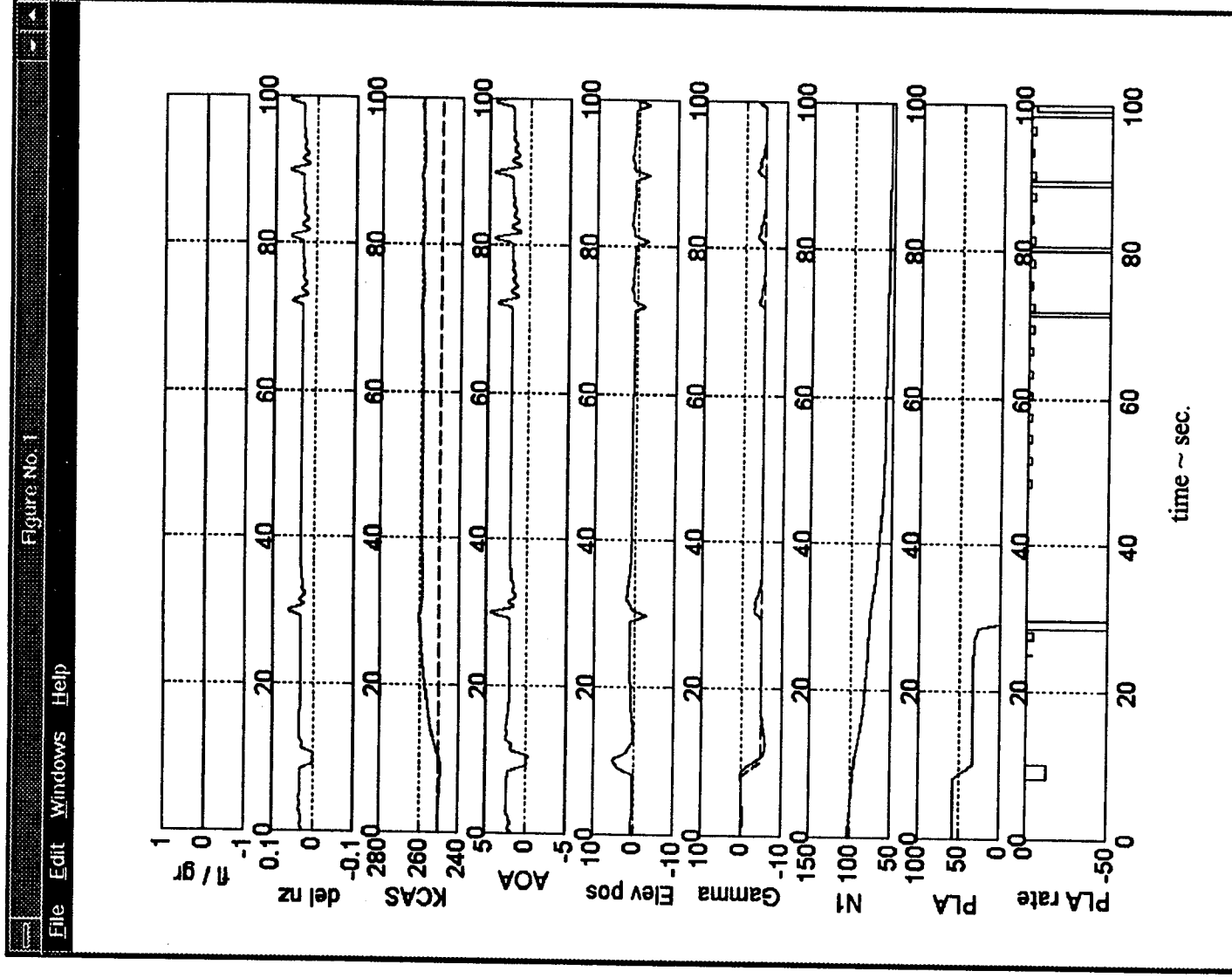


Figure 25. High altitude high speed and overspeed protection time history for the jet. See figure 31 (page 64) for legend.

Figure 26 shows a time history of the business jet exercising the stall protection feature. The airplane was trimmed for level flight at 150 knots. The controllers were turned on at 2 seconds. A level flight command at a speed slower than the level flight stall speed was given. As the airplane slowed the angle of attack increased until the maximum allowed of 12 degrees was reached at about 40 seconds. At this time the stall protection feature was activated and the elevator moved to reduce the angle of attack to less than 12 degrees. At the same time, the throttle moved rapidly to full power so as to minimize altitude loss. The airplane then entered a series of oscillations centered about the  $\alpha$  limit while accelerating and descending until sufficient speed was recovered to allow level flight, at which time the throttle started to move back in another attempt to achieve the commanded speed. Since the actual stall angle of attack for this airplane is 20 degrees, an actual stall did not occur even during the oscillations. The presence of this oscillatory behavior suggests that the elevator controller might benefit from a circuit that adapts the elevator movement to airplane response similar to the one incorporated in the speed controller.



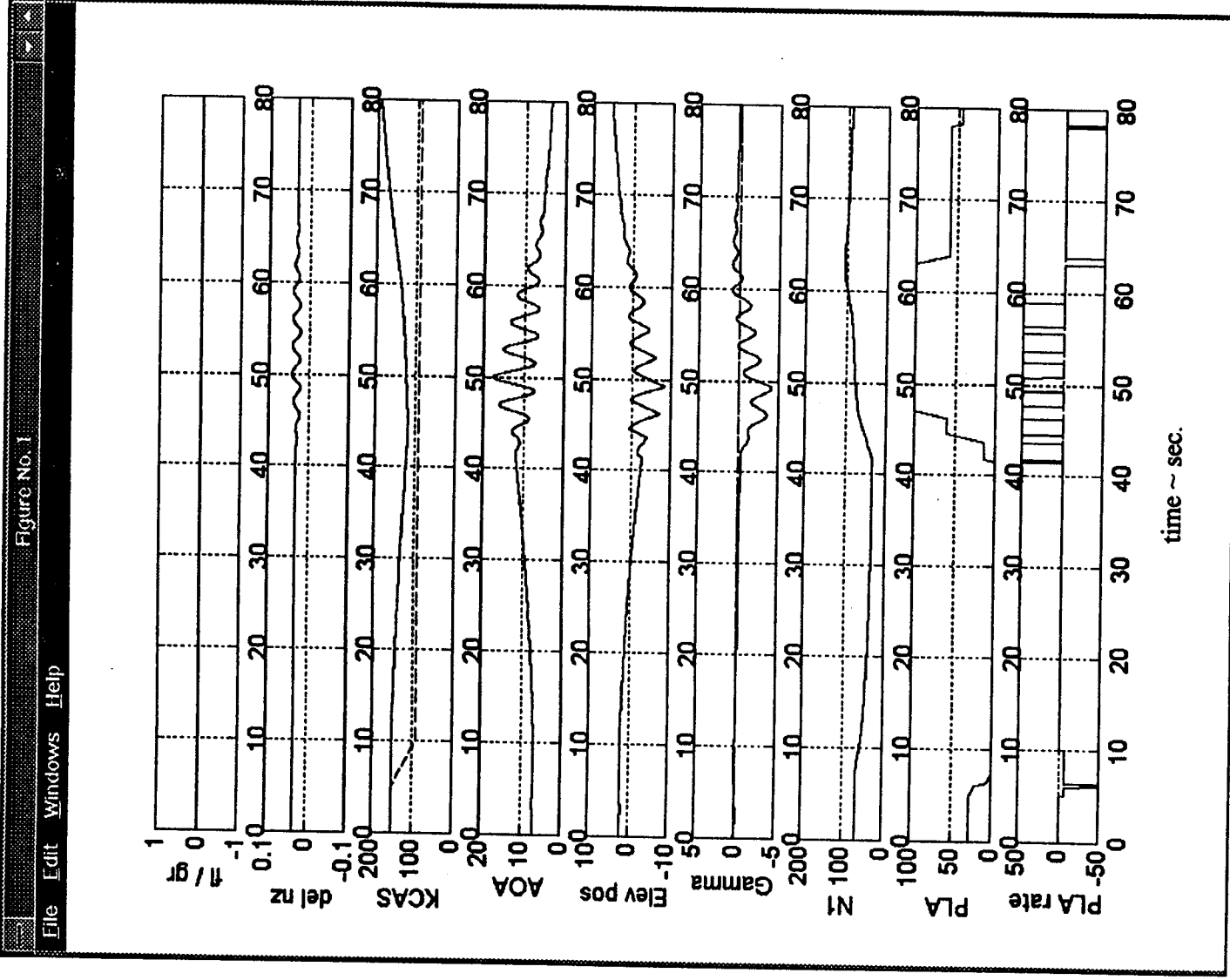


Figure 26. Stall protection time history for the jet.

See figure 31 (page 64) for legend.

## **5.2 Piston Single Simulation Results**

Figure 27 shows the time history of the piston simulation transitioning from slow cruise to final approach and then executing a missed approach in turbulence. The simulation was started with the airplane trimmed for level flight at 130 knots. At 2 seconds the controllers were turned on. At 10 seconds an airspeed of 90 knots was commanded. At 50 seconds the gear is lowered. At 60 seconds a flight path of 3 degrees down was commanded. At 100 seconds the flaps are extended. At 120 seconds a missed approach was initiated by commanding a flight path angle of 3 degrees up. This case demonstrated the characteristics of the controllers coupled with the piston airplane at medium to low speed with configuration changes.

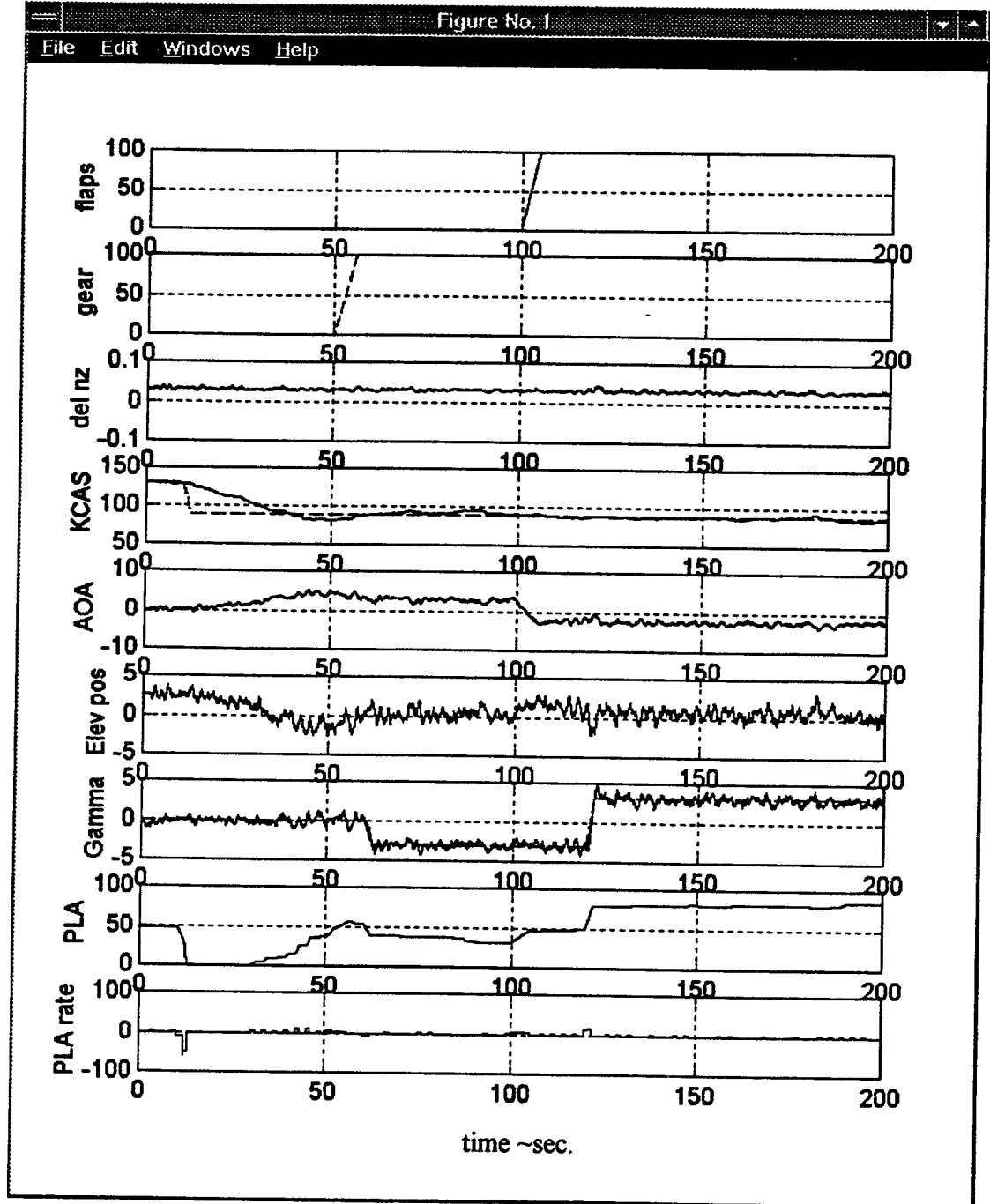


Figure 27. Approach and miss time history for the piston airplane.

See figure 31 (page 64) for legend.

Figure 28 shows a time history of the piston powered single engine airplane climbing at maximum power, then leveling and accelerating to maximum level flight speed. The airplane was initially trimmed at 130 knots with power for level flight. The controllers are turned on at 2 seconds and command an airspeed of 180 knots with a climb angle of 3 degrees. The throttle moved to 100 percent while the flight path followed the gamma command. At 50 seconds the airplane had stabilized at the maximum speed it can attain while maintaining the commanded flight path. At 75 seconds a level flight path was commanded. The airplane followed this command and accelerated to maximum level flight speed (which is less than the commanded speed). When the level off command was given, the throttle moved from the maximum position momentarily. This is a nuisance area where more intelligence could have been built into the predictive throttle movement circuit. However, it was thought that specific knowledge of an airplane's maximum flight speed would jeopardize the generality of this system.

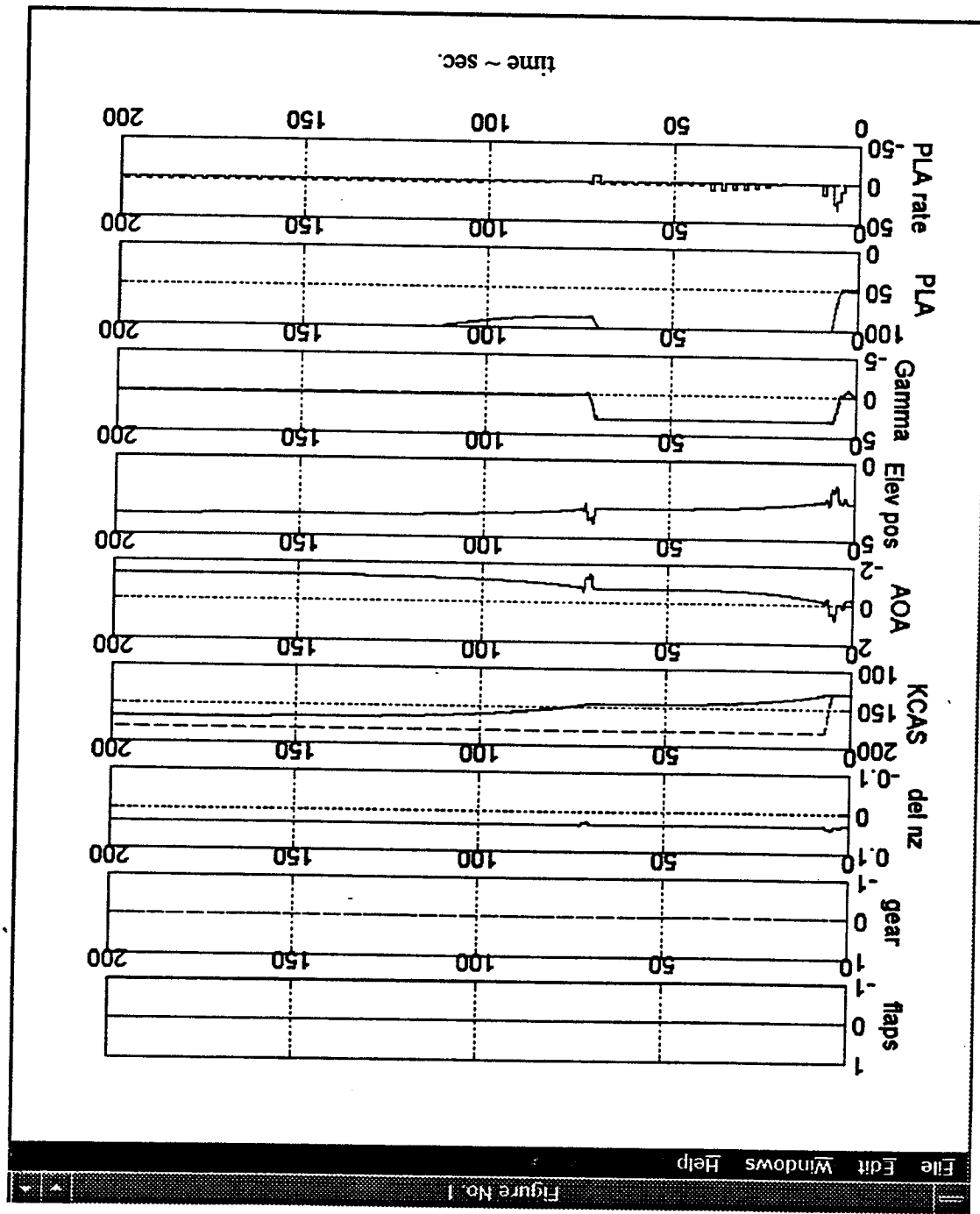


Figure 29 shows a time history of the piston powered single engine airplane exercising the overspeed protection feature. With the airplane trimmed for level flight at 130 knots, the controllers were turned on and commands given to accelerate to the maximum allowable speed and dive at 10 degrees. The steep decent was commanded so that a sufficient acceleration could be maintained such that an overspeed would occur. At about 55 seconds the overspeed did occur. At this time the throttle moved rapidly to almost idle. As the airspeed stabilized the throttle moved forward to maintain the commanded speed.

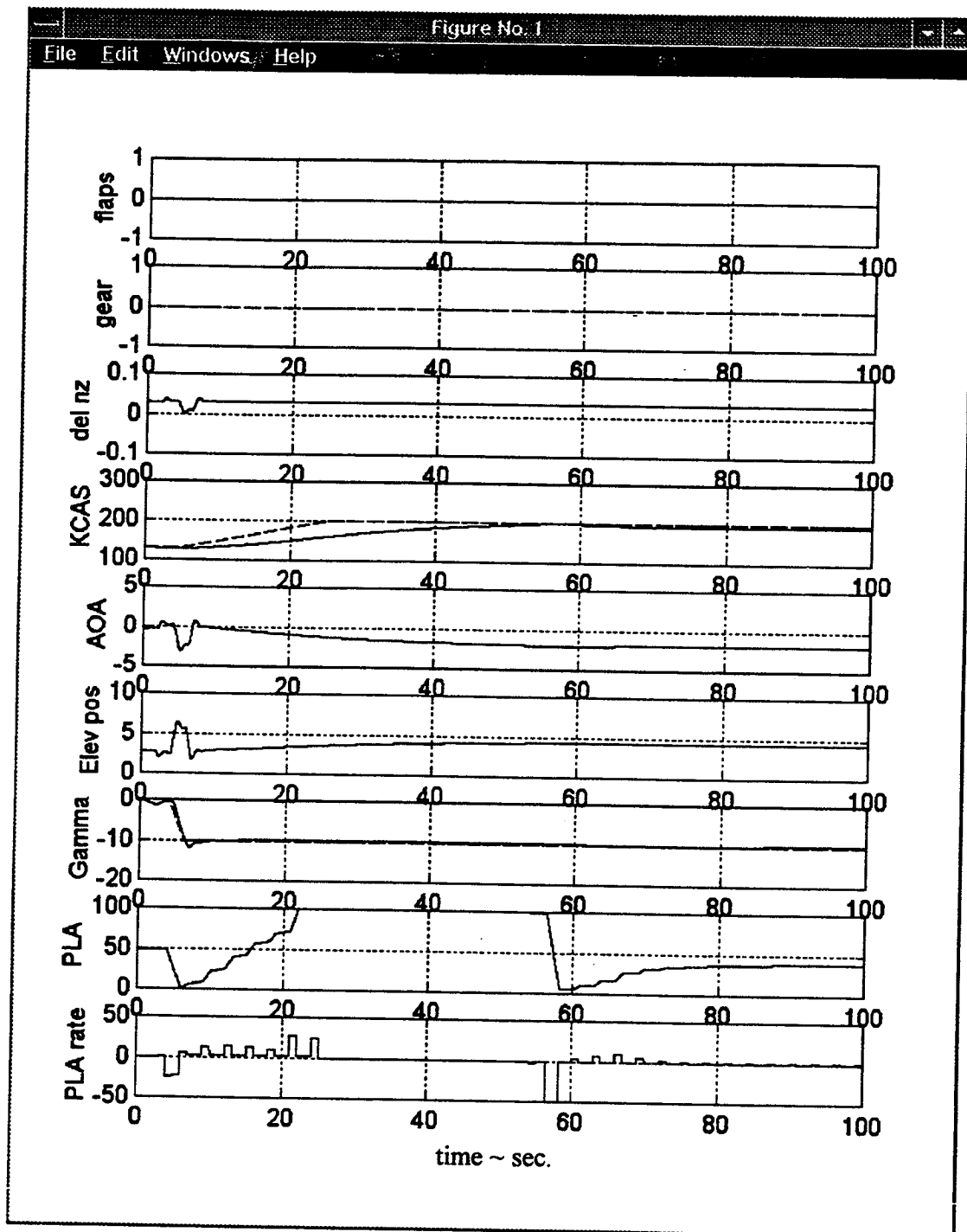


Figure 29. Overspeed protection time history for the piston airplane.

See figure 31 (page 64) for legend.

Figure 30 shows a time history of the piston powered single engine airplane exercising the stall protection feature. The airplane was set up in level flight at 130 knots. The controllers were turned on at 2 seconds. A level flight path was commanded at a speed slower than the level flight stall speed. As the airplane slowed the angle of attack increased until the maximum allowed of 12 degrees was reached at 54 seconds. At this time the stall protection feature was activated and the elevator moved to reduce the angle of attack to less than 12 degrees. At the same time, the throttle moved rapidly to increase power so as to minimize altitude loss. The throttle and elevator controllers then worked together to maintain an angle of attack near the maximum while holding close to the commanded flight path.



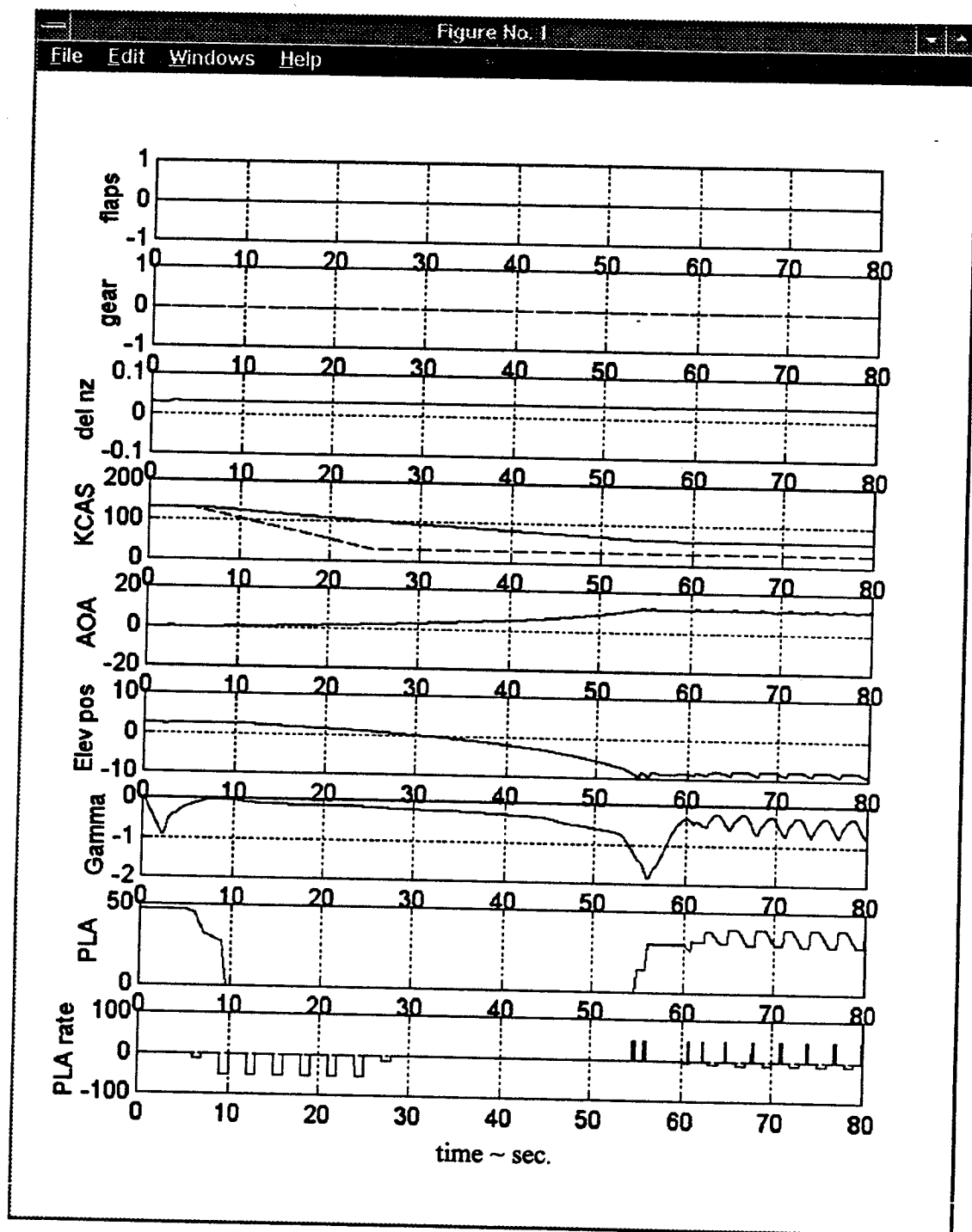


Figure 30. Stall protection time history for the piston single.

See figure 31 (page 64) for legend.

Plot Legend	
flaps (fl)	Flap position 0 is retracted, 100% is full down (solid line for jet)
gear (gr)	Landing gear position 0 is retracted 1 is extended (dashed line for jet)
del nz	Change in normal acceleration in gs from 1g
KCAS	Knots Calibrated Airspeed (dashed line is commanded)
AOA	Angle of attack in degrees
Elev pos	Elevator position in degrees
PLA rate	Power lever rate , percent per second
Gamma	Flight path angle in degrees (dashed line is commanded)
PLA	Power lever angle, percent
N1	Jet engine fan speed 104% is maximum allowable

Figure 31. Legend for the plot labels of figures 23 - 30

## **6 DISCUSSION**

### **6.1 Unexpected Discoveries**

There were several unexpected discoveries associated with this project. These in general had to do with the psychology of flying an airplane and fact that the development of the controllers naturally followed the same progression as a human does when learning to fly.

#### **6.1.1 Difficulty Obtaining Accurate and Complete Knowledge**

One of the most difficult tasks in developing a fuzzy model of a human controlling a complex machine is that of extracting information that is both complete and accurate from a human expert. It was expected that since this project required extracting this knowledge from the author (who holds an airline transport pilot license and is an instrument flight instructor) that this task would be relatively simple (no miscommunication, immediate access, etc.). Given this expectation, it was a surprise to discover the amount of difficulty involved in accurately analyzing the reaction and control processes that are required to fly an airplane. Many of the required actions are in fact trained reflexes which are not easily transferred from the subconscious to the conscious part of the brain. Even when this transfer was successfully made, the form the information took was usually in a graphical or tactile form, not in the form of set boundaries and output rules. For example, a combination of visual images received by the pilot triggers a learned reflex of applying a given amount of force (governed by tactile feedback) to the control column. The task then became one of converting a visual image into fuzzy input sets, and learned reflexes into rules and output sets with which a computer can work.

This difficulty shed some light on the reasons why learning to fly an airplane well in instrument conditions requires such a large investment in time and effort both for initial training and recurrent proficiency.

#### **6.1.2 Instrument Pilots Operate Like Timesharing Digital Devices**

A considerable amount of effort was spent attempting to get reasonable engine response from the jet engine simulation. There was a high confidence level that the fuzzy input sets, rules and outputs were correct or nearly so, but the aircraft's speed response to command changes was not satisfactory. After much introspection and discussions with other pilots, it was discovered that pilots (especially in instrument conditions) behave like timesharing digital devices with time split unevenly between tasks.

If systems management is ignored, the pilot who is flying an ILS (precision approach using only aircraft instruments) has three basic things to control, and one input to monitor closely. The control requirements are vertical flight path, horizontal flight path and airspeed. The monitored input is altitude (which triggers a missed approach). In accomplishing this task, the pilot is trained to continuously move his eyes in a predefined pattern from one instrument to another without stopping at any one instrument longer than it takes to interpret the data displayed by that instrument. He then formulates and executes a control input based on that data while moving to the next instrument. Response to that control is then received and evaluated the next time he scans that instrument. For the airspeed indicator, the scanning cycle takes from 3 to 5 seconds for an experienced pilot. The pilot gets many more indications of vertical flight path (altimeter, glide slope, vertical speed indicator, g loading, etc.) and therefore controls it with an update rate that can be faster than one second.

Even though the continuous flight path controller was acceptable, a 0.05 second zero order sample and hold was incorporated to model a digital design.

A 3 second zero order sample and hold was implemented in the throttle controller. This improved the airplane response somewhat, but was still unacceptable. It was then realized that although this sample time was about right, the pilot does not move the throttle continuously during this time period. The output logic block was added at this time to limit throttle travel to the second immediately following the taking of the sample. This worked well with the piston simulation, but the jet simulation went into non-linear oscillations during some flight conditions.

### **6.1.3 How Pilots Adapt to Engine Response Characteristics**

One of the differences between piston powered airplanes and corporate jets is the engine's response to throttle changes. A normally aspirated piston engine with a constant speed propeller restabilizes on a new power setting almost immediately, while a jet engine takes several seconds to restablize. In addition, the jet engine accelerates much more slowly than it decelerates. These engine characteristics in turn cause these two airplanes to have very different speed response characteristics as a result of a throttle change by the pilot. The human pilot adapts to this by thinking further ahead of the airplane (anticipating) and making earlier power changes in the jet. He also knows that the jet engine will not respond to rapid throttle changes, and therefore makes slow changes. (This technique also works for the piston airplane, but is not required.)

Since the controller developed in this project was intended to be used on both types of engines, some scheme of adapting the controller to the engine was required, or the controller would need to be made to act slowly enough that it could accommodate any engine. Since piston powered airplanes generally have faster responses to

turbulence, and therefore sometimes need faster throttle movements to control speed, it was decided that an adaptive controller was the best type.

Since pilots use engine response to determine the maximum allowable frequency of throttle changes, this same type of system was incorporated into the electronic controller. The engine response was determined by continuously monitoring the acceleration rate. The maximum effective frequency was governed by not allowing a subsequent throttle change until the engine had nearly stabilized from the previous change. Also, since pilots have the ability to determine that a larger change is required than was first estimated and then command that change without waiting for the engine to stabilize, this feature was also incorporated into this controller.

#### **6.1.4 Controller Development Compared to Human Learning**

The development sequence used during this project was:

- 1) Fuzzy flight path controller without the sample and hold feature
- 2) Fuzzy speed controller
- 3) Sample and hold input for the speed controller
- 4) Pulsed output for the speed controller
- 5) Add sample and hold, and output logic to flight path controller
- 6) Turbulence detector and corresponding gain reduction
- 7) Predictive power changes
- 8) Envelope protection

It is interesting to note that if items 2, 3, 4 and 5 are grouped together, that this represents a typical sequence of learning for a human who is in the process of earning a private pilot license.

When the beginning student is given his first task (control flight path) by itself, he tends to behave like a continuous control device. However, when the second function

is added (accurate speed control) he tends to fixate on one or the other, thus causing a pronounced version of the digital response characteristics described above. Thus, items 2, 3, 4 and 5 all develop simultaneously in the human pilot.

Even though there was no conscious attempt for controller development to follow the sequence normally experienced by flight students, it was interesting to note that for this project where mirroring human behavior in the final product was desired, that this was the sequence that occurred.

It is generally accepted that a business jet is more difficult to fly than a piston powered single engine airplane. These controllers did a better job controlling the piston airplane than the jet airplane, despite the fact that they were developed on the jet. These controllers were developed by attempting to translate a human pilot's knowledge into computer software. Given this, is it possible that the knowledge translation was accurate and complete enough that this ease of controlling bias toward the smaller airplane got transferred to the controllers?

## **6.2 Continued Research**

Since this is a new control scheme, there are areas where improvements can be made in performance. This is the first area for continued work.

The next step is to expand it to include the lateral axis. Also, if this type of control scheme is to be used throughout the entire flight envelope, then takeoff and landing must be considered. After these features are implemented, takeoff and landing during gusty cross winds needs to be addressed. And in the final analysis, to ensure that this type of control scheme really works, it needs to be installed on an airplane and flight tested.

Since there is some latitude in the control parameters, an area of continued research would be to determine how sensitive the final results are to each of these parameters. Also, to determine which parameters have the largest effects.

Because analyzing time histories is an inefficient way to design control systems, there is a real need to develop better tools for designing controllers for non-linear systems in general and fuzzy logic controllers in particular.



## REFERENCES

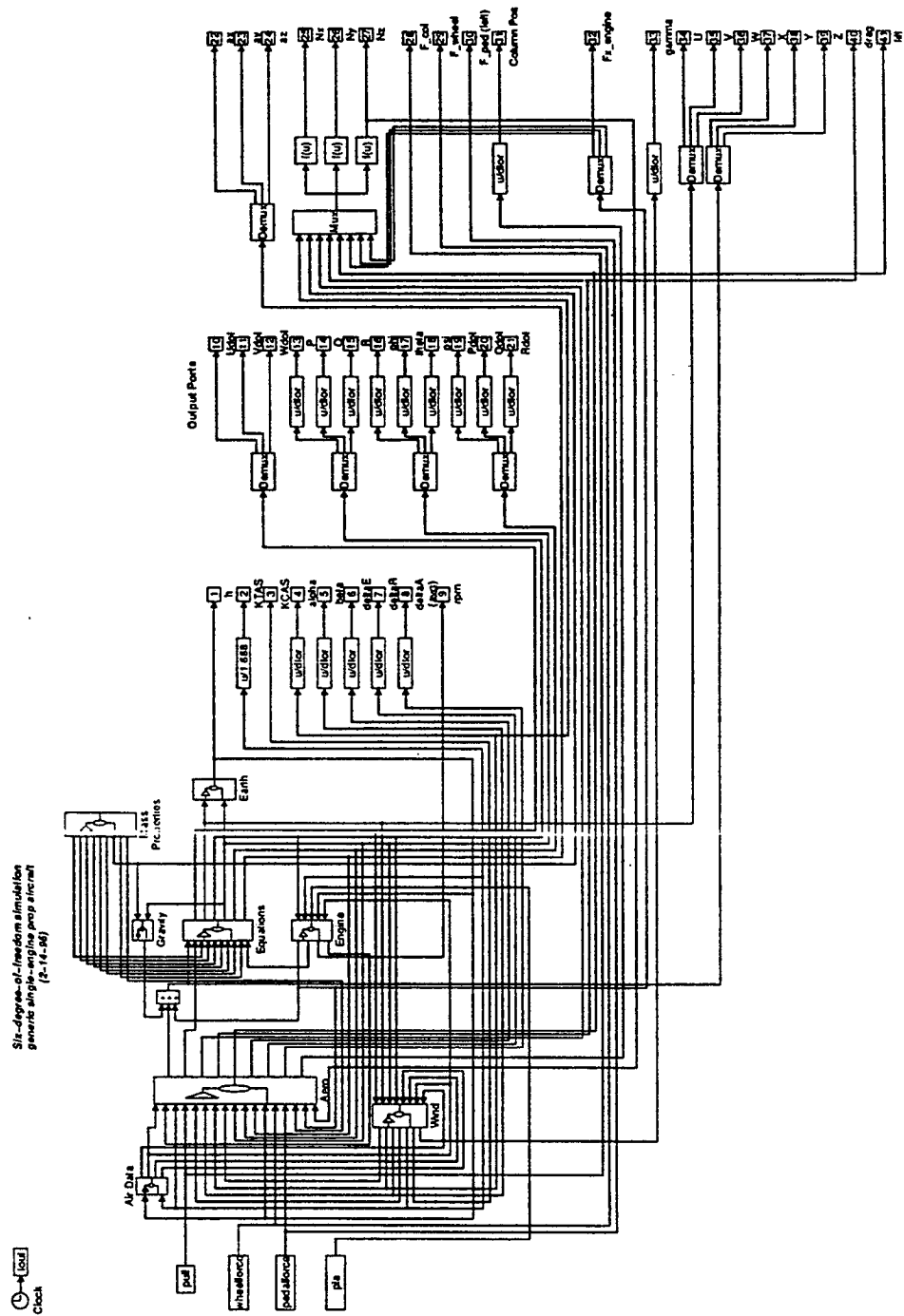
- [1] University of Kansas-Flight Research Laboratory, "Final Report on the Flight Test Program for a Separate Surface Stability Augmented Beech Model 99" KU-FRL report #364, The University of Kansas, Lawrence Ks. 66045, for Dryden Flight Research Center NASA, Contract No. NAS 4-2148, 1976
- [2] Feinreich, Benjamin; Seckel, Edward; Ellis, David R.; "In Flight Simulation Study of Decoupled Longitudinal Controls for the Approach and Landing of a STOL Aircraft", NASA CR-2710, 1977
- [3] Stewart, E. C.; Ragsdale, W. A.; Wunschel, A. J., "An evaluation of automatic control system concepts for general aviation airplanes" AIAA paper 88-4364, 1988
- [4] Stewart, E. C., "A piloted simulation study of advanced flight controls and displays for novice general aviation pilots", AIAA paper 94-0276, 1994
- [5] Hwang, W. R.; Tao, C. W.; Thompson, W. E.; Paz, R.; Design methodology for fuzzy controllers and comparison with an optimal PD controller for a nonlinear control system", Midwest Symposium on Circuits and Systems v 1 1993. Publ by IEEE, IEEE service center Piscataway NJ., pp 25-28, 1993
- [6] Brehm, Thomas; Rattan, Kuldip S.; "Hybrid fuzzy logic PID controller", 1993 National Aerospace and Electronics Conference Proceedings, pp 807-813, 1993
- [7] Mamdani, E. H.; "Twenty years of fuzzy control. Experiences gained and lessons learnt", 1993 IEEE International Conference on Fuzzy Systems, pp 339-344, 1993
- [8] Reinfrank, Michael; "Fuzzy control systems. Clear Advantages" Siemens Review v 58 n 6 Nov-Dec 1991 pp 28-32, 1991
- [9] Berenji, Hamid R.; "Fuzzy and neural control", NASA-TM-108753, May 1992
- [10] Steinberg, Marc; "Potential role of neural networks and fuzzy logic in flight control design and development", AIAA paper 92-0999, 1992

- [11] Steinberg, Marc L.; Digirolamo, Robert D.; "Neural network and fuzzy logic technology for naval flight control interim report", Report No. AD-A242650; NADC-91080-60, Naval Air Development Center, Warminster, Pa., Air Vehicle and Crew Systems Technology Dept., Available from the AIAA Technical Library, 1990
- [12] Gonsalves, Paul G.; Zacharias, Greg L., "Fuzzy logic gain scheduling for flight control", IEEE International Conference on Fuzzy Systems, Vol. 2, 1994
- [13] Layne, Jeffery R.; Passino, Kevin M., "Fuzzy model reference learning control for cargo ship steering", IEEE Control Systems Magazine, Vol. 13, No. 6, pp 23-34, December 1993
- [14] Berenji, Hamid R.; Lea, Robert N.; Jani, Yashvant; Khedkar, Pratap; Malkanj, Anil; Hoblit, Jeffery, "Space shuttle attitude control by reinforcement learning and fuzzy logic", 1993 IEEE International Conference on Fuzzy Systems Second IEEE International Conference on Fuzzy Systems, 1993
- [15] Chiu, Stephen; Chand, Sujeet, "Fuzzy controller design and stability analysis for an aircraft model", Proceedings of the American Control Conference, Vol. 1, pp 821-826, IEEE cat. no. 91CH2939-7, 1991
- [16] Zhang, Shuguang; "Fuzzy control for roll agility", ICAS, Congress, 19th, Anaheim, Ca., Sept. 18-23, 1994, Proceedings. Vol. 3 (A94-13534 10-01), American Institute of Aeronautics and Astronautics, pp 2296-2301, 1994
- [17] Baker, Craig, "Fuzzy logic control in flight control systems", Fuzzy logic control - Software and hardware applications (Prentice Hall Series on Environmental and Intelligent Systems. Vol. 2) pp 363-373, 1993
- [18] Sugeno, M.; Nishino, J.; Miwa, H., "Fuzzy control of an unmanned helicopter", Proceedings of the Third International Workshop on Neural Networks and Fuzzy Logic, Volume 1, p 107, January 1993
- [19] Steinberg, Marc; "A fuzzy logic based F/A-18 automatic carrier landing system", AIAA Guidance, Navigation and Control Conference, Hilton Head Island, SC, Aug 10-12, 1992, Technical Papers Pt. 1, pp 407-417, Washington, American Institute of Aeronautics and Astronautics, 1992

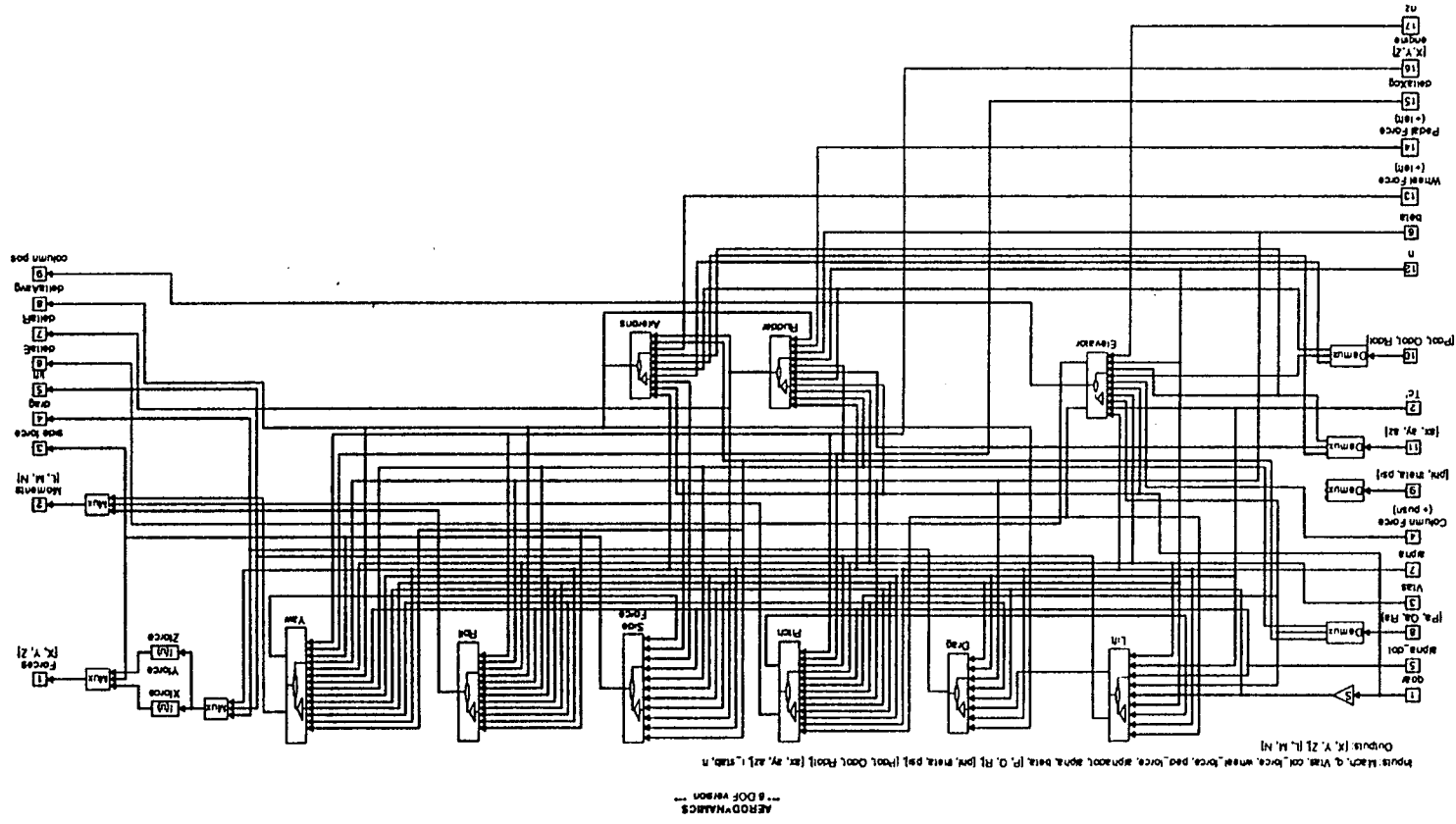
- [20] Wood, David; Schneider, Moti, "Fuzzy expert system control for naval helicopters", IEEE International Conference on Fuzzy Systems, Vol. 2, 1994
- [21] Zadeh, L.A., "Fuzzy sets," Information and Control, Vol. 8, pp.338-353, 1965
- [22] The Mathworks Inc., "Fuzzy Logic Toolbox For Use With Matlab"  
24 Prime Park Way, Natick, Mass. 01760, 1995
- [23] Gupta, Madan M.; Saridis, George N.; Gaines, Brian-R., Fuzzy Automata and Decision Processes, North Holland Publishing Co., Netherlands, 1977
- [24] Pedrycz, Witold, Fuzzy Control and Fuzzy Systems Second, Extended, Edition, Wiley & Sons, 1993
- [25] Mamdani, E. H.; Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp 1-13, 1975
- [26] Sugeno, M., Industrial applications of fuzzy control,  
Elsevier Science Pub. Co., 1985
- [27] Pfeiffer, Bernard-Markus; Isermann, Rolf, "Criteria for successful applications of fuzzy control", Engineering Applications of Artificial Intelligence v 7 n 3 Jun 1994, p 245-253, 1994

## **Appendix**

### **Simulink Block Diagrams of the Piston Powered Single Engine Airplane**

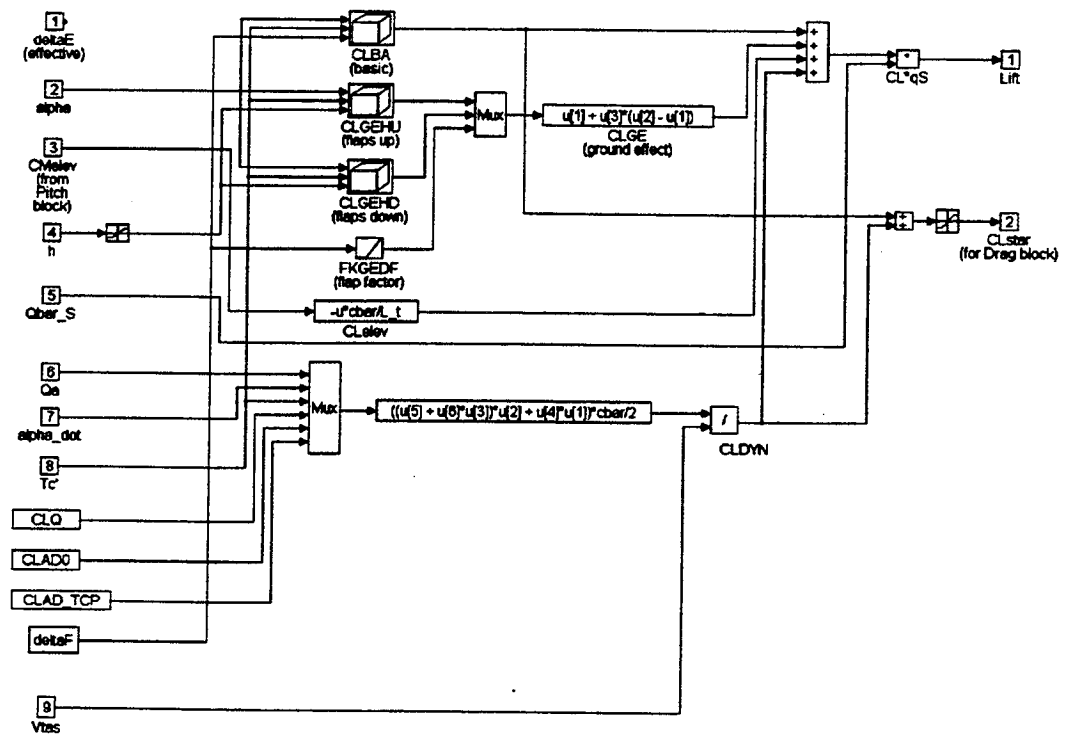


Simulink piston powered single engine airplane block diagram



## LIFT

LOOKUP TABLES MUST USE DEGREES

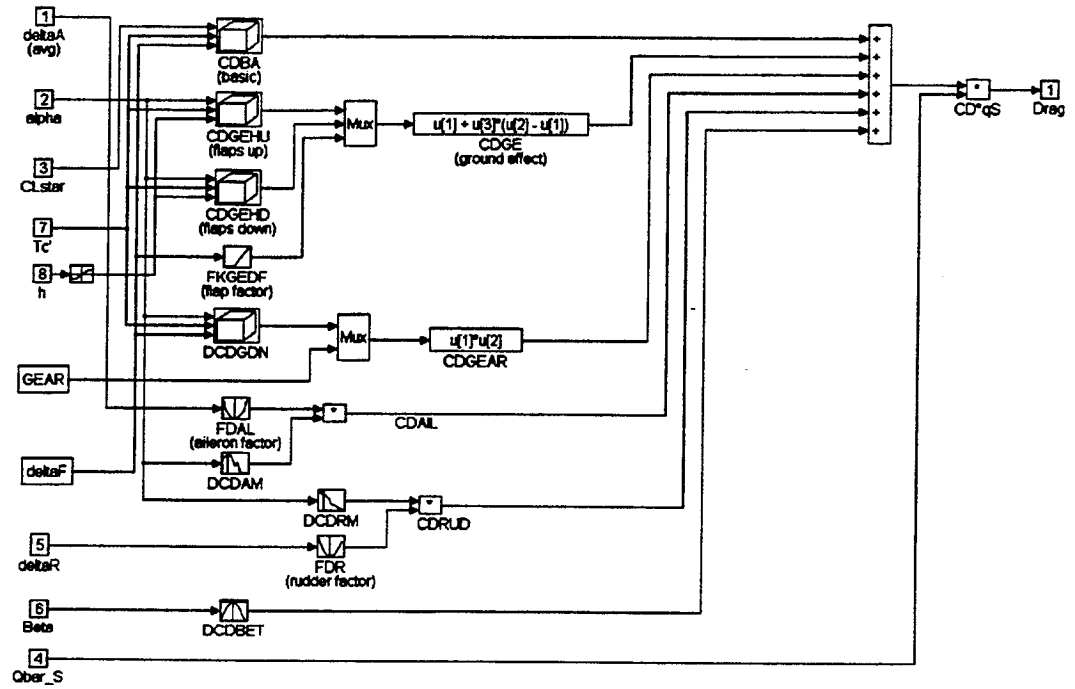


Lift subsystem



## DRAG

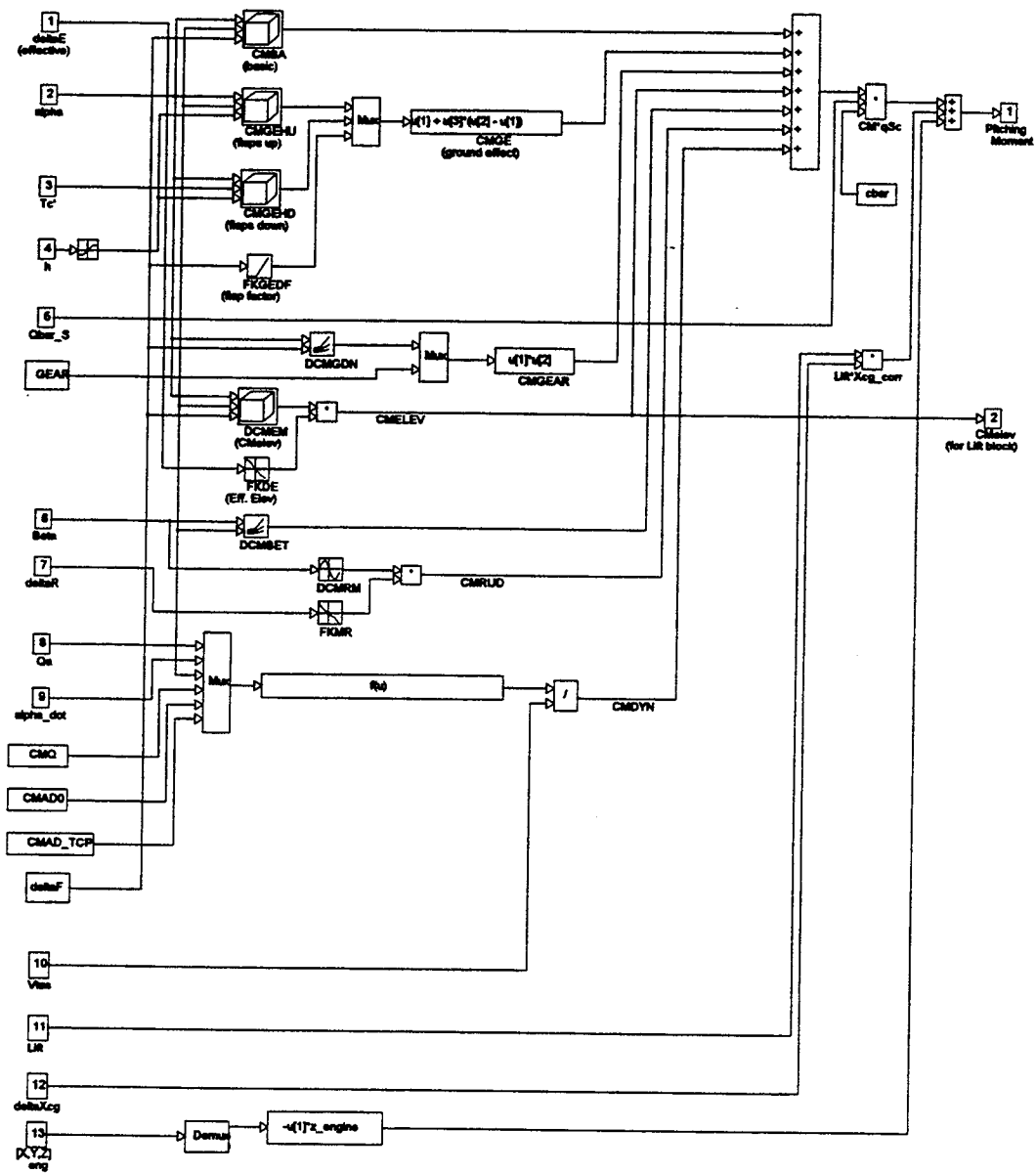
LOOKUP TABLES MUST USE DEGREES



Drag subsystem

### PITCHING MOMENT

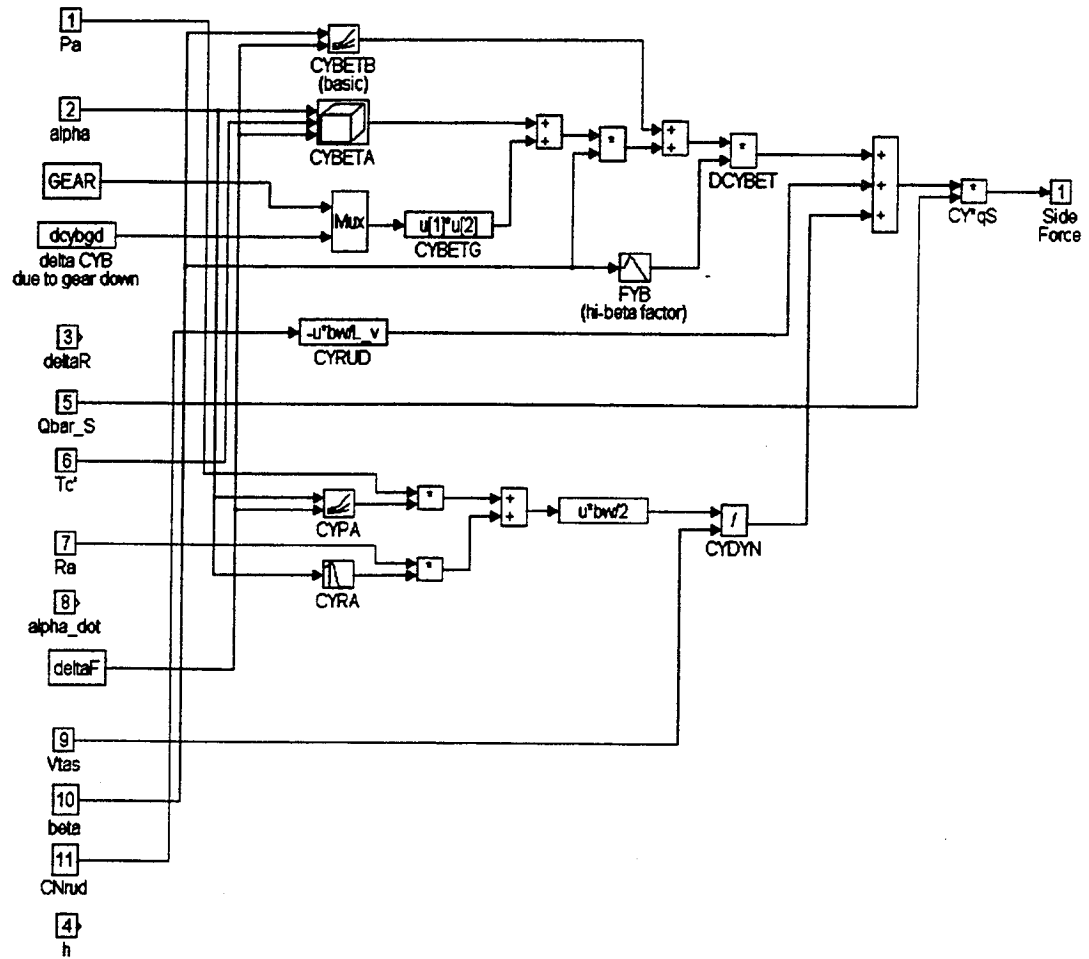
**LOOKUP TABLES MUST USE DEGREES**



### Pitching moment subsystem

## SIDE FORCE

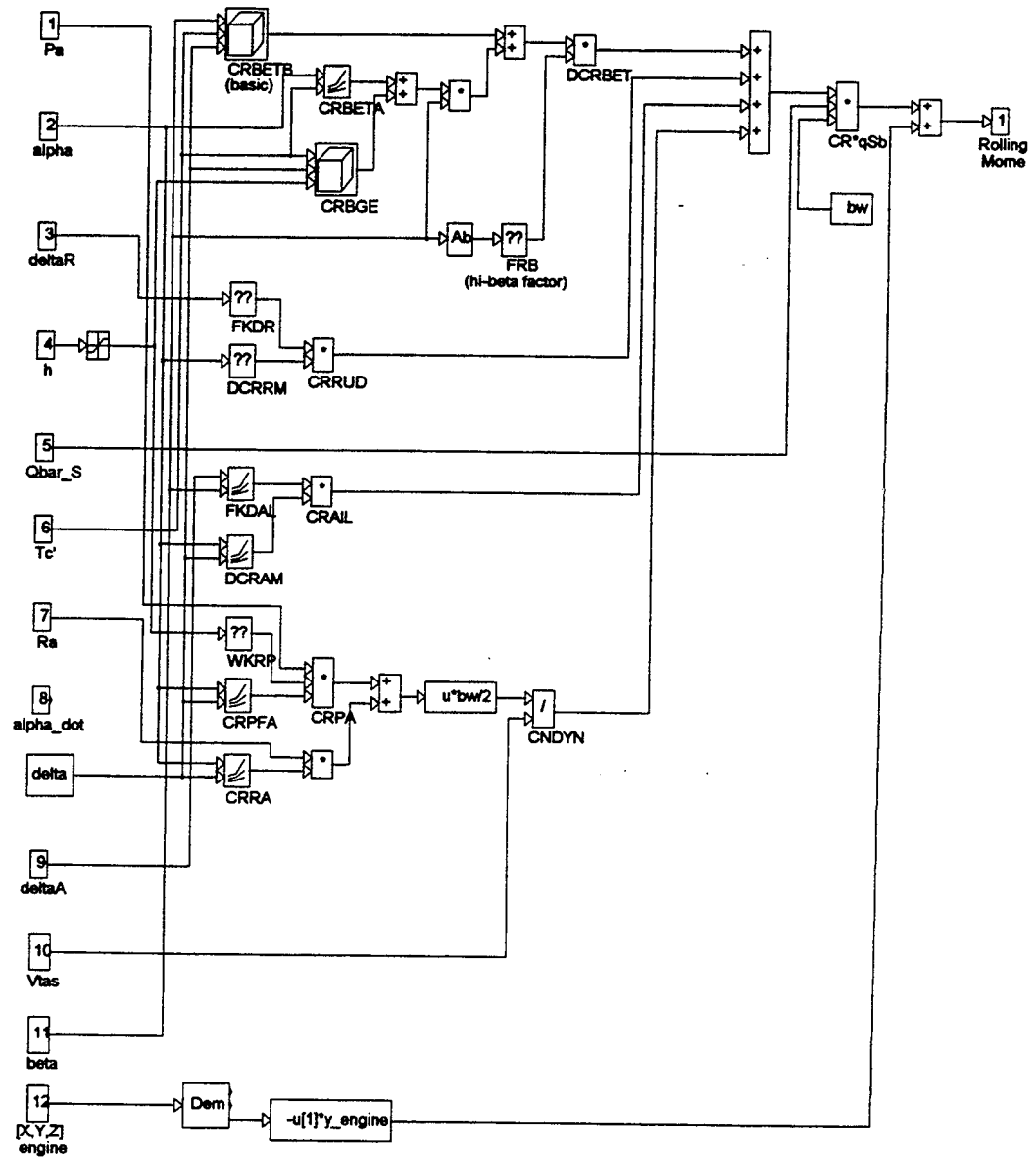
LOOKUP TABLES MUST USE DEGREES



Sideforce subsystem

## ROLLING MOMENT

LOOKUP TABLES MUST USE DEGREES



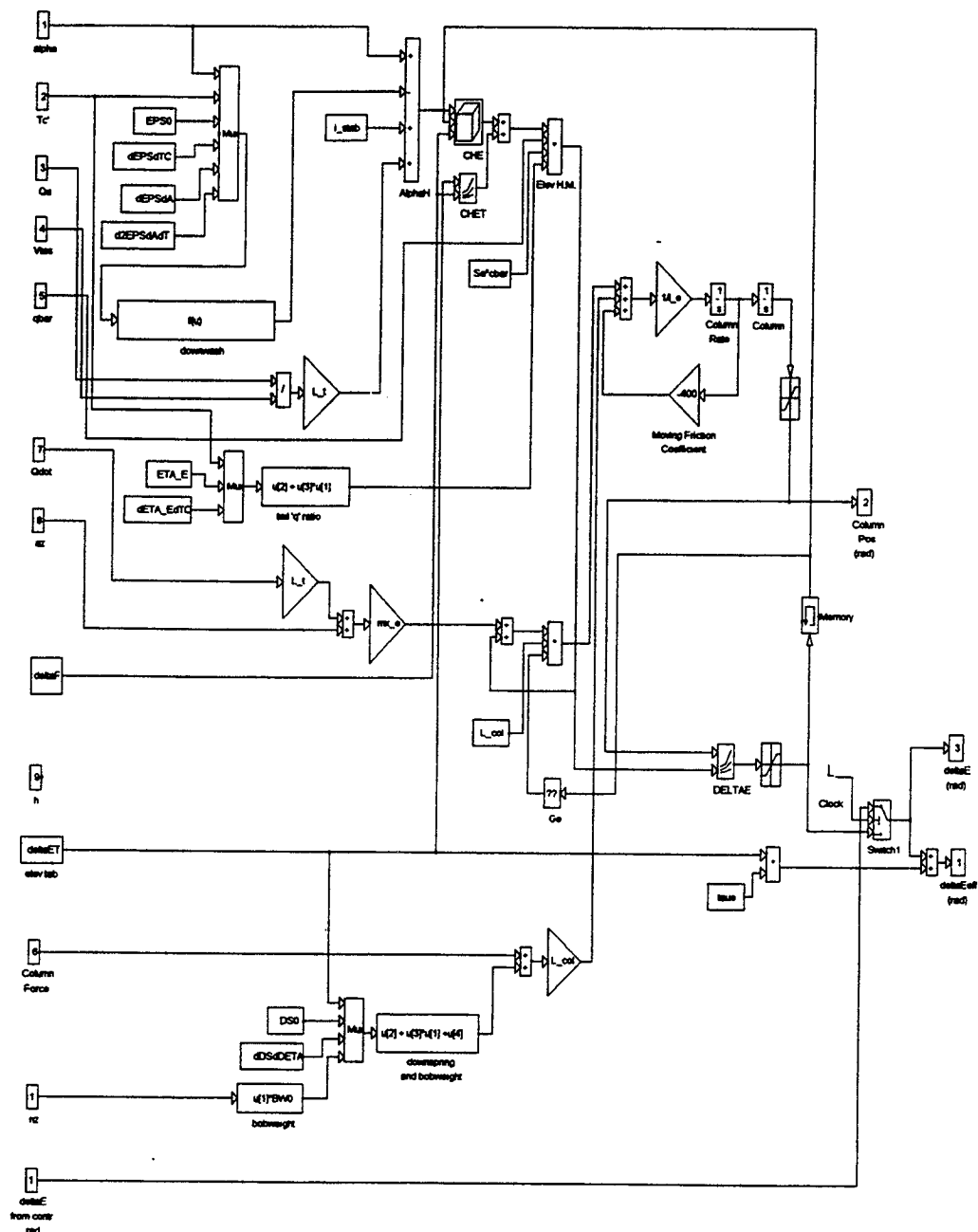
Rolling moment subsystem

The diagram illustrates a control system for a ship's yawing motion. It features a reference input  $\dot{\alpha}$  (labeled 'alpha\_dot') which is integrated to produce a reference yaw angle  $\alpha$  (labeled 'alpha'). This reference angle is compared with the actual yaw angle (labeled 'beta') to generate an error signal. This error signal is processed through a series of blocks, including a gain block  $K$  and a feedback loop involving the actual yaw rate  $\dot{\beta}$  (labeled 'beta\_dot'). The resulting signal is then used to calculate the yawing moment, which is the final output of the system. The diagram also shows various intermediate signals and feedback loops, such as the yaw rate feedback loop and the yaw angle feedback loop, which are used to stabilize the ship's motion.

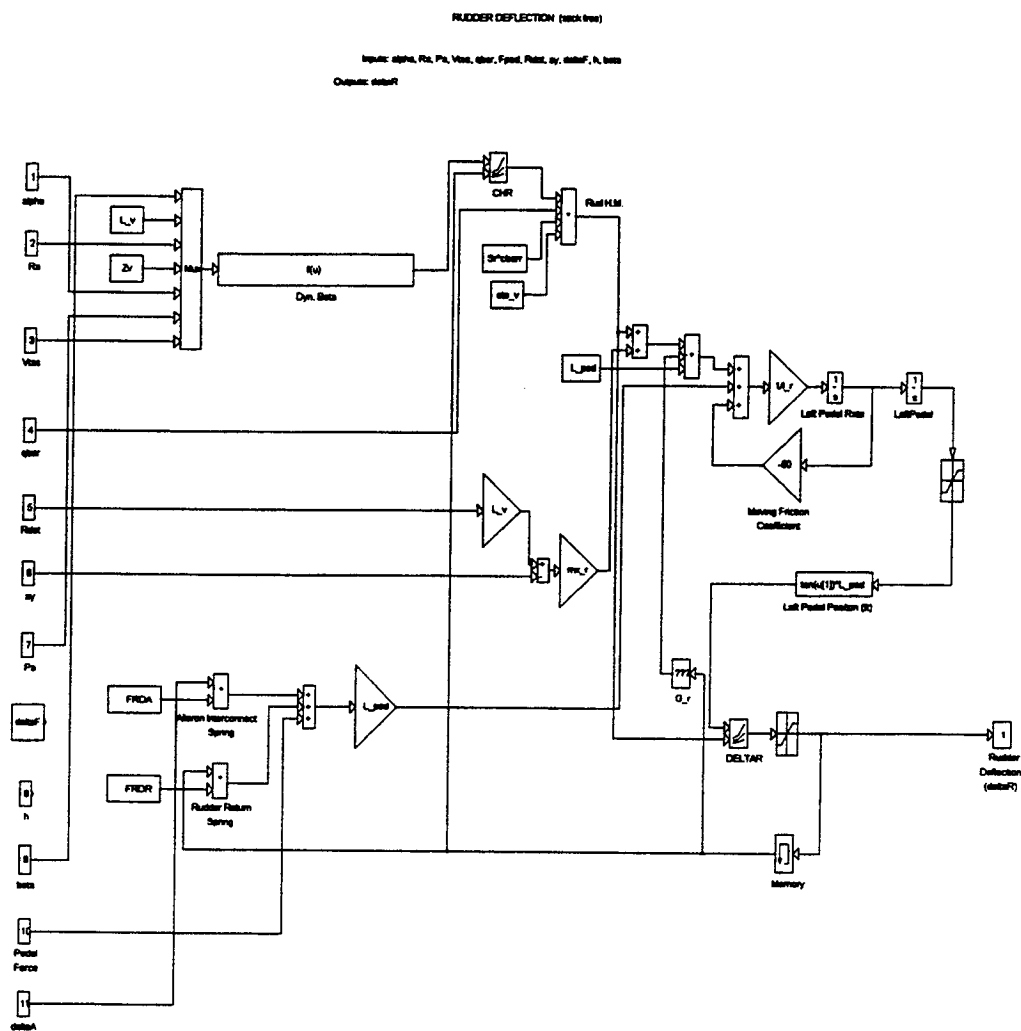
### Yawing moment subsystem

Input: alpha, Qa, Vmax, gbar, Foot, Qdot, az, Mach, deltaF, h

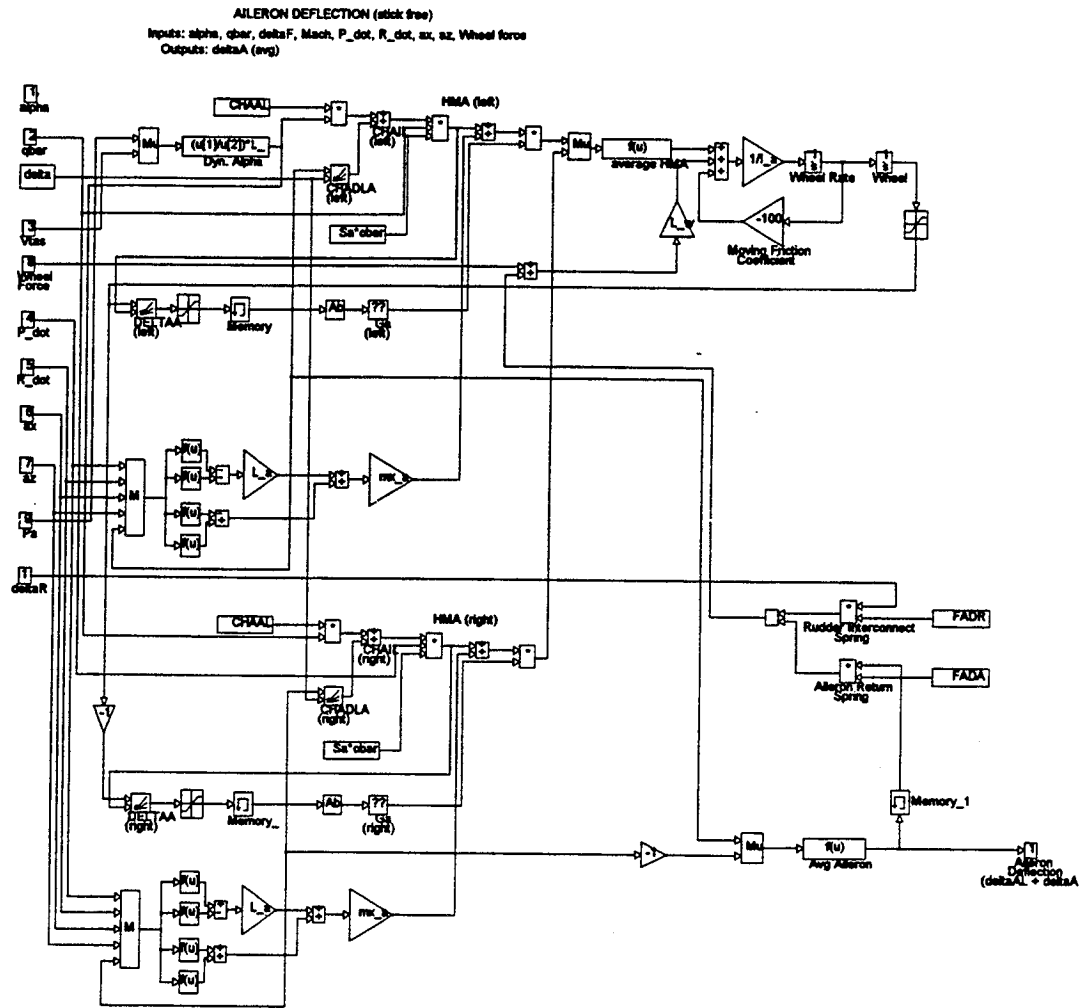
**Output device**



### Elevator deflection subsystem



### Rudder deflection subsystem



Aileron deflection subsystem

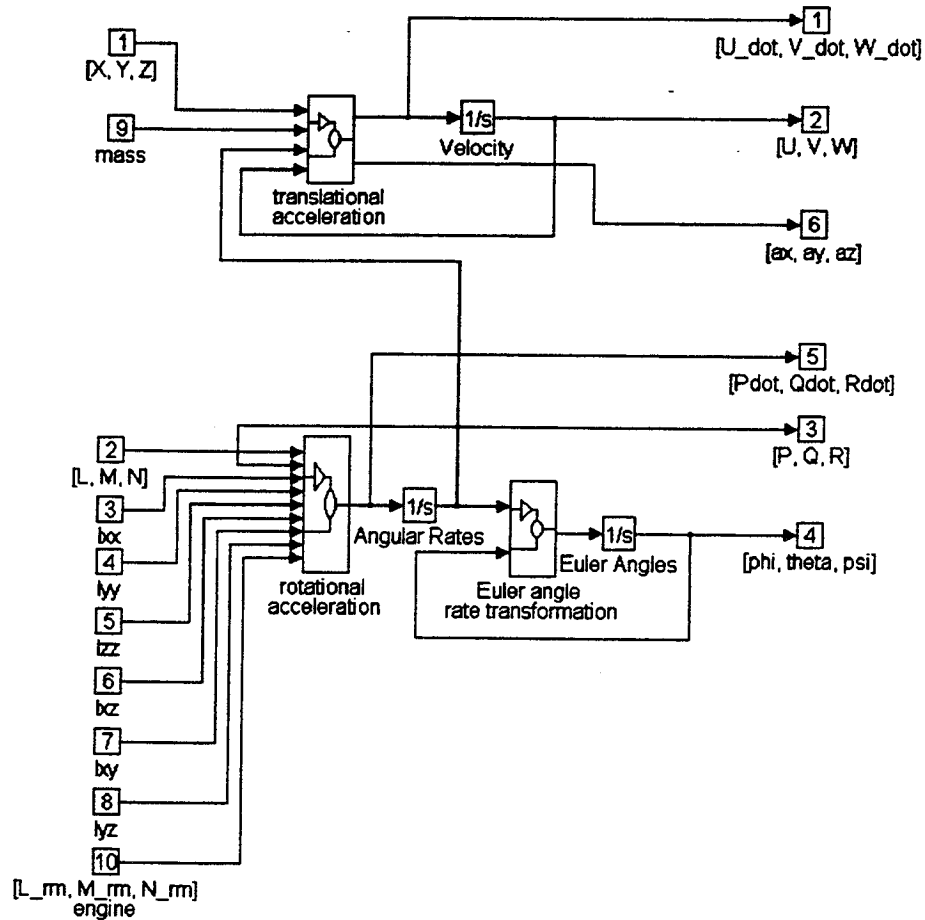


### EQUATIONS OF MOTION

This block generates the linear acceleration, linear and angular velocities, and the Euler angles from the forces, moments, mass and inertias

Inputs:  $[X, Y, Z]$ ,  $[L, M, N]$ ,  $b_x, l_{yy}, l_{zz}, b_z, b_y, l_{yz}$ , mass,  $[L_{rm}, M_{rm}, N_{rm}]$

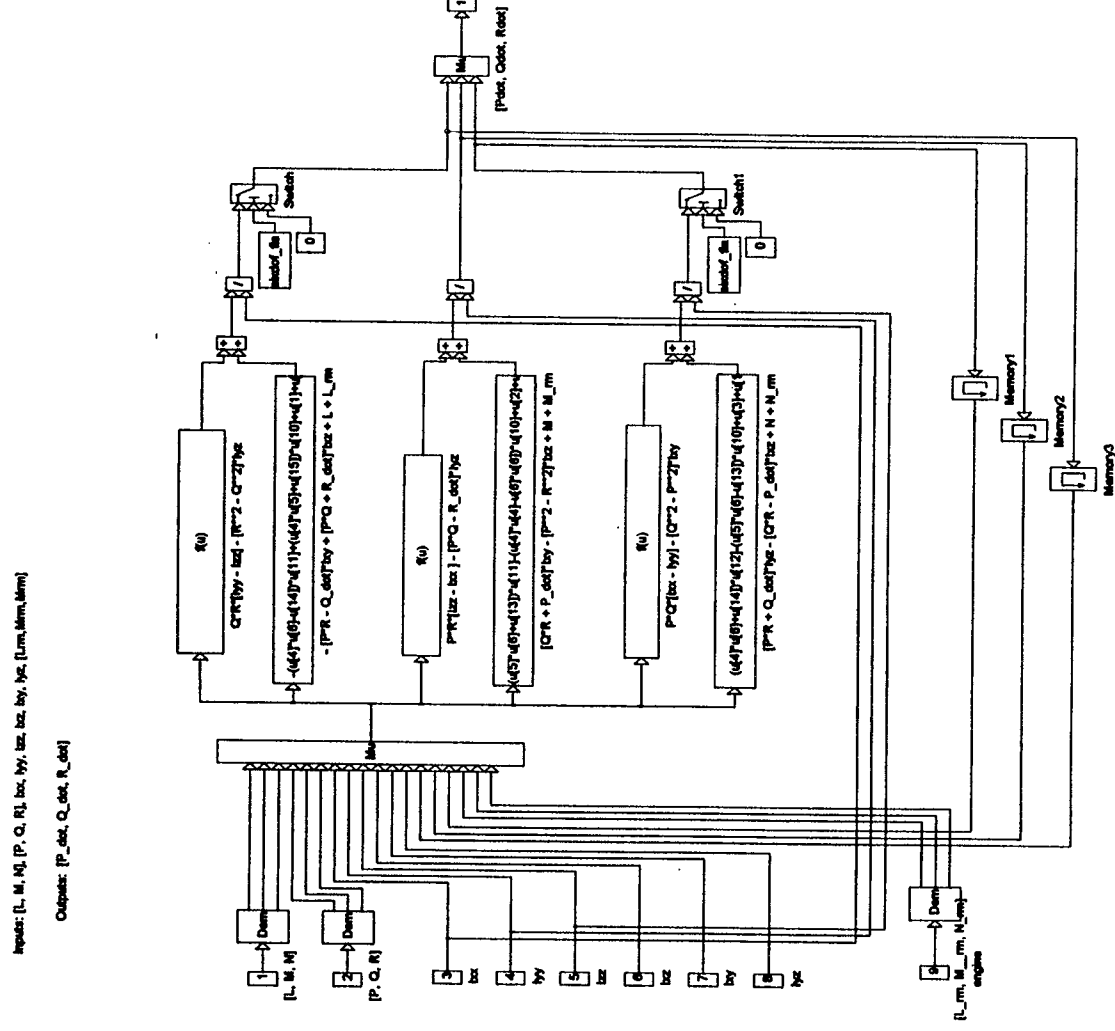
Outputs:  $[U_{dot}, V_{dot}, W_{dot}]$ ,  $[U, V, W]$ ,  $[P, Q, R]$ ,  $[\phi, \theta, \psi]$ ,  $[P_{dot}, Q_{dot}, R_{dot}]$ ,  $[a_x, a_y, a_z]$ ,  $V_{tas}$



Equations of motion subsystem

## ROTATIONAL ACCELERATION

**This block generates the rotational partial acceleration from the moments and inertias and the angular rates**



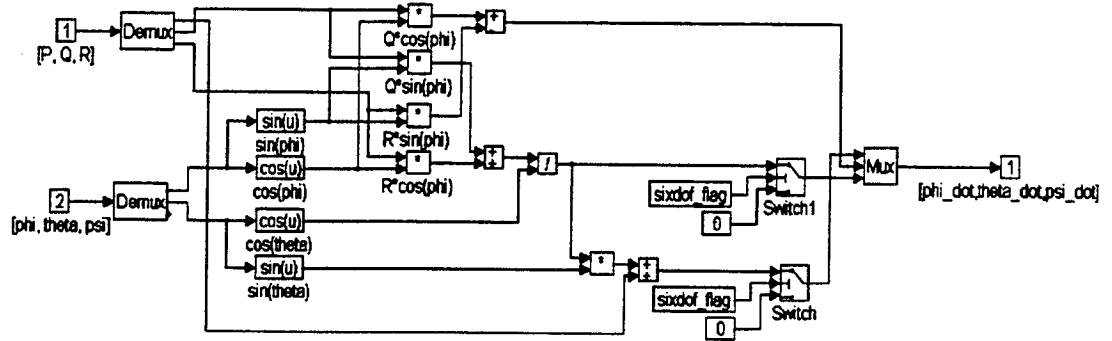
### Rotational acceleration subsystem

### EULER ANGLE RATE TRANSFORMATION

This block generates the Euler angle rates from the body angular rates and the Euler angles

Inputs:  $[P, Q, R]$ ,  $[\phi, \theta, \psi]$

Outputs:  $\phi_{\dot{}}$ ,  $\theta_{\dot{}}$ ,  $\psi_{\dot{}}$



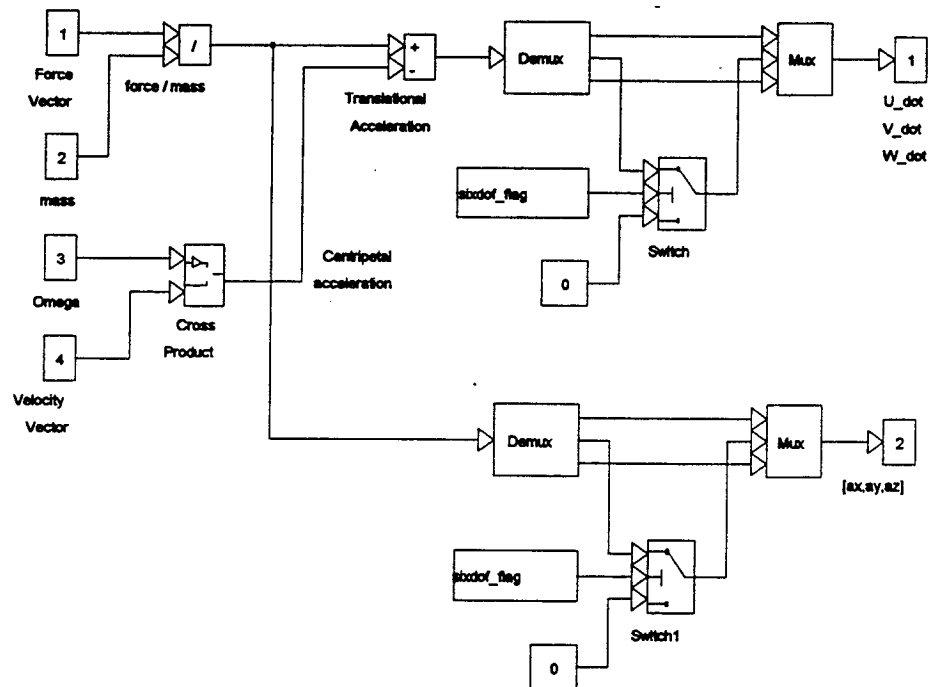
Euler angle rate transformation subsystem

# TRANSLATIONAL ACCELERATION

This block determines the translational acceleration from the linear force, the mass, and the angular and linear velocities

Inputs: [X, Y, Z], mass, [P, Q, R], [U, V, W]

Outputs: [U\_dot, V\_dot, W\_dot]

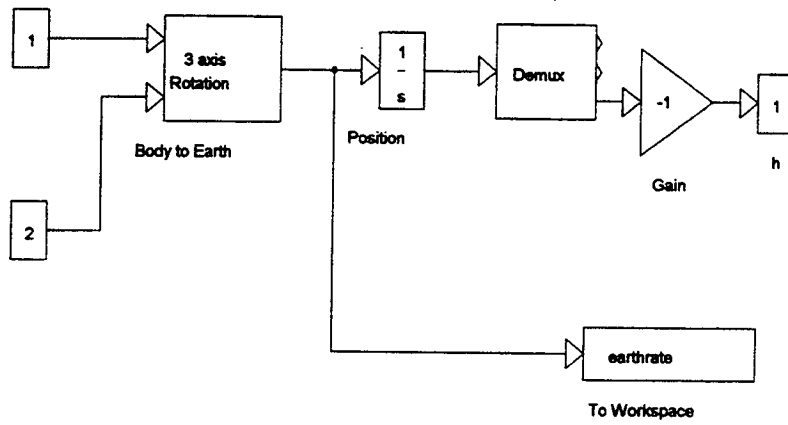


Translational acceleration subsystem

## EARTH

Inputs: [U, V, W], [phi, theta, psi]

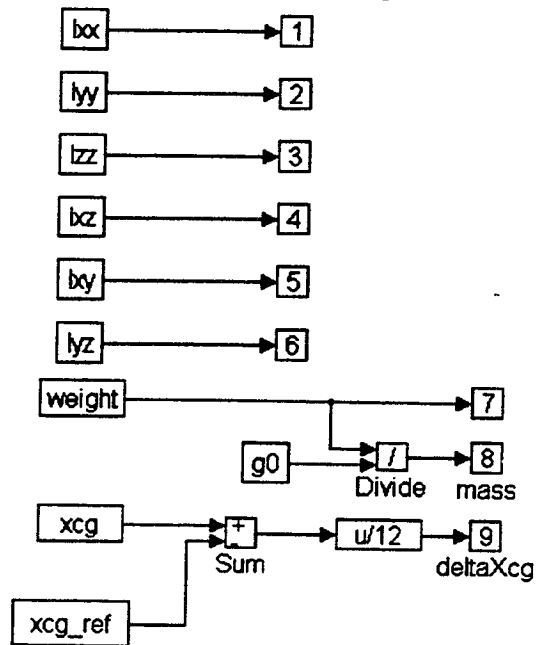
Outputs: [x\_earth, y\_earth, z\_earth], h



## Earth subsystem

**MASS PROPERTIES**

Outputs: bx, ly, lzz, bz, by, lyz, mass, weight

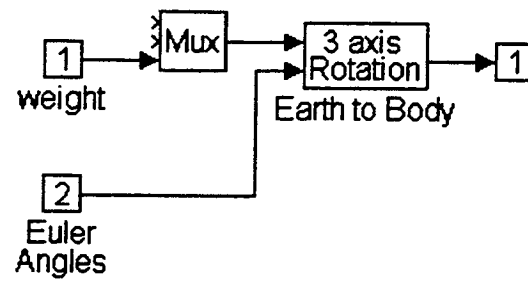


Mass properties subsystem

## GRAVITY

Inputs: weight, [phi, theta, psi]

Outputs: [Xg, Yg, Zg]

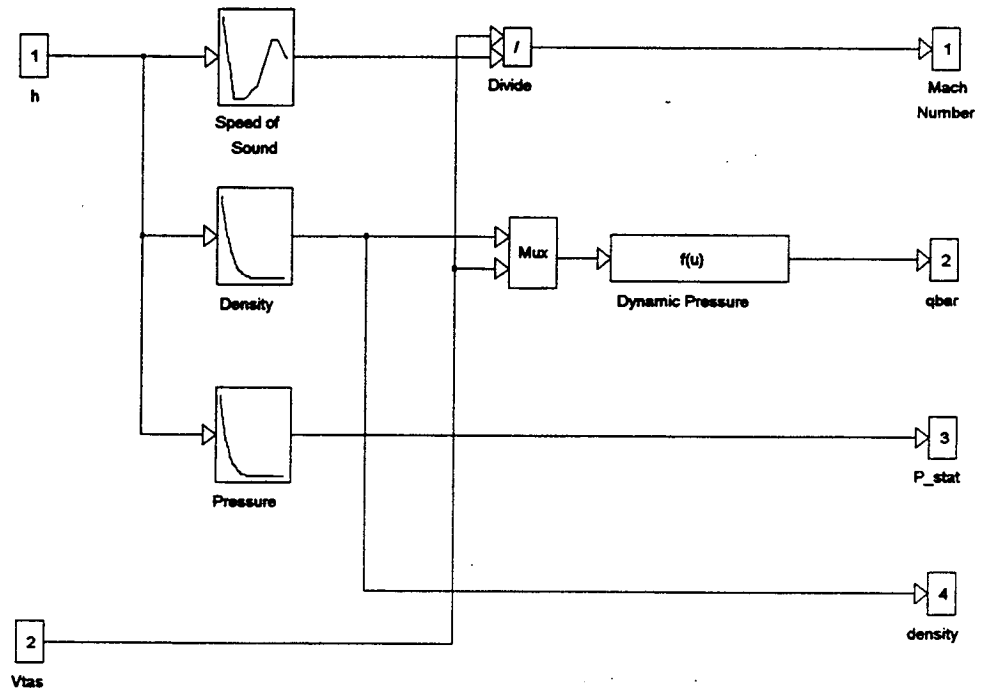


Gravity subsystem

# AIR DATA

Inputs: h, Vtas

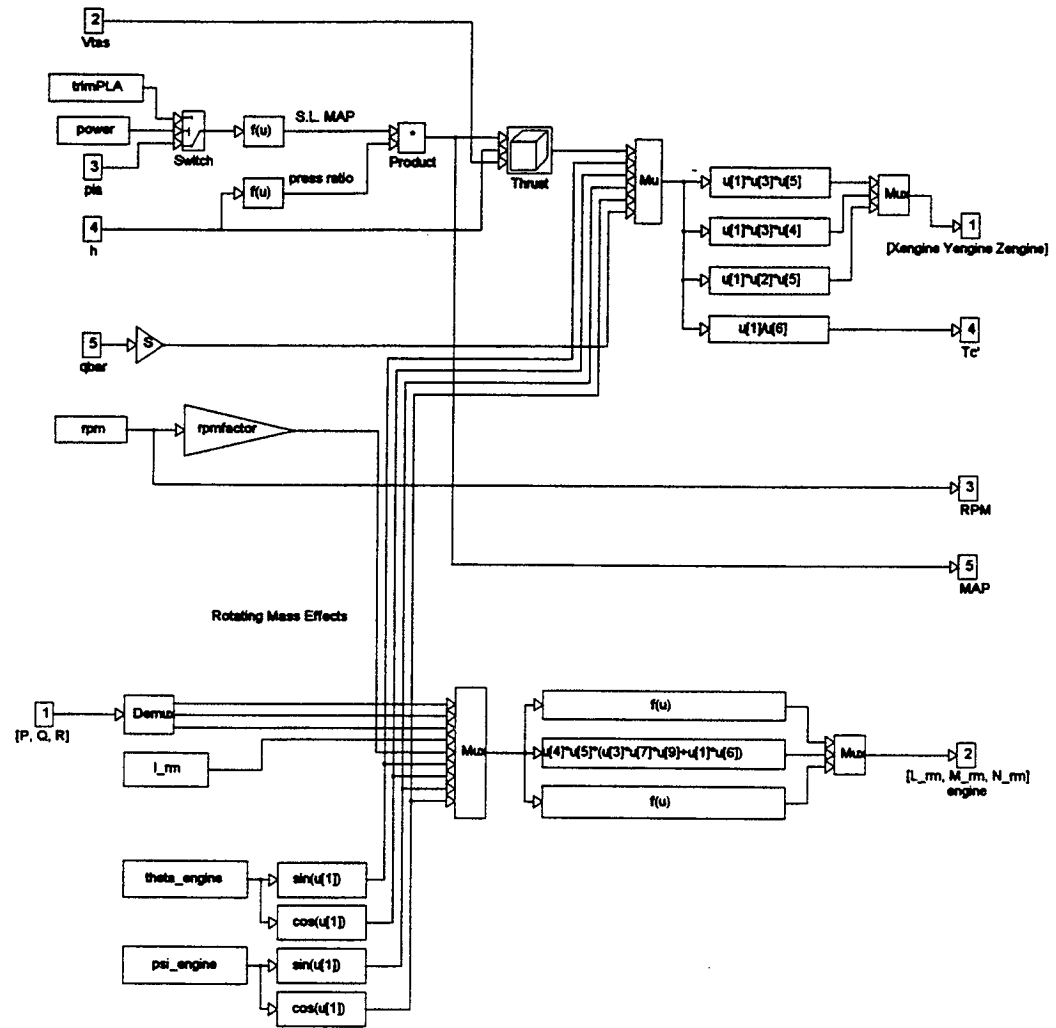
Outputs: Mach q



Air data subsystem



NORMALLY ASPERATED  
RECIPROCATING ENGINE  
WITH CONSTANT SPEED PROP

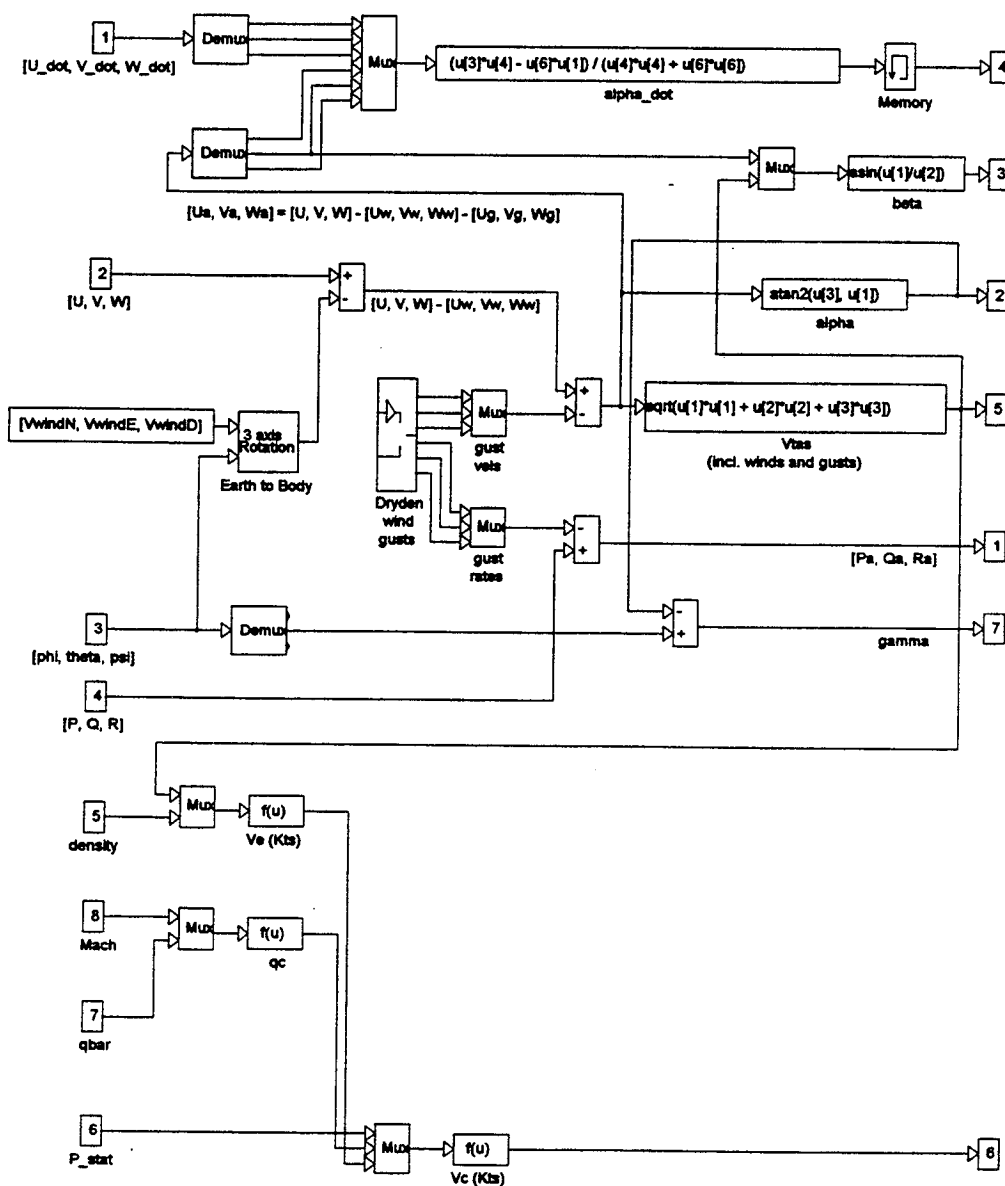


Engine subsystem

## WIND

This block generates the angles of attack and sideslip and the angle of attack rate from the inertial linear velocity and acceleration, and the wind and gust velocities. The wind velocity is transformed to body coordinates using the Euler angles.

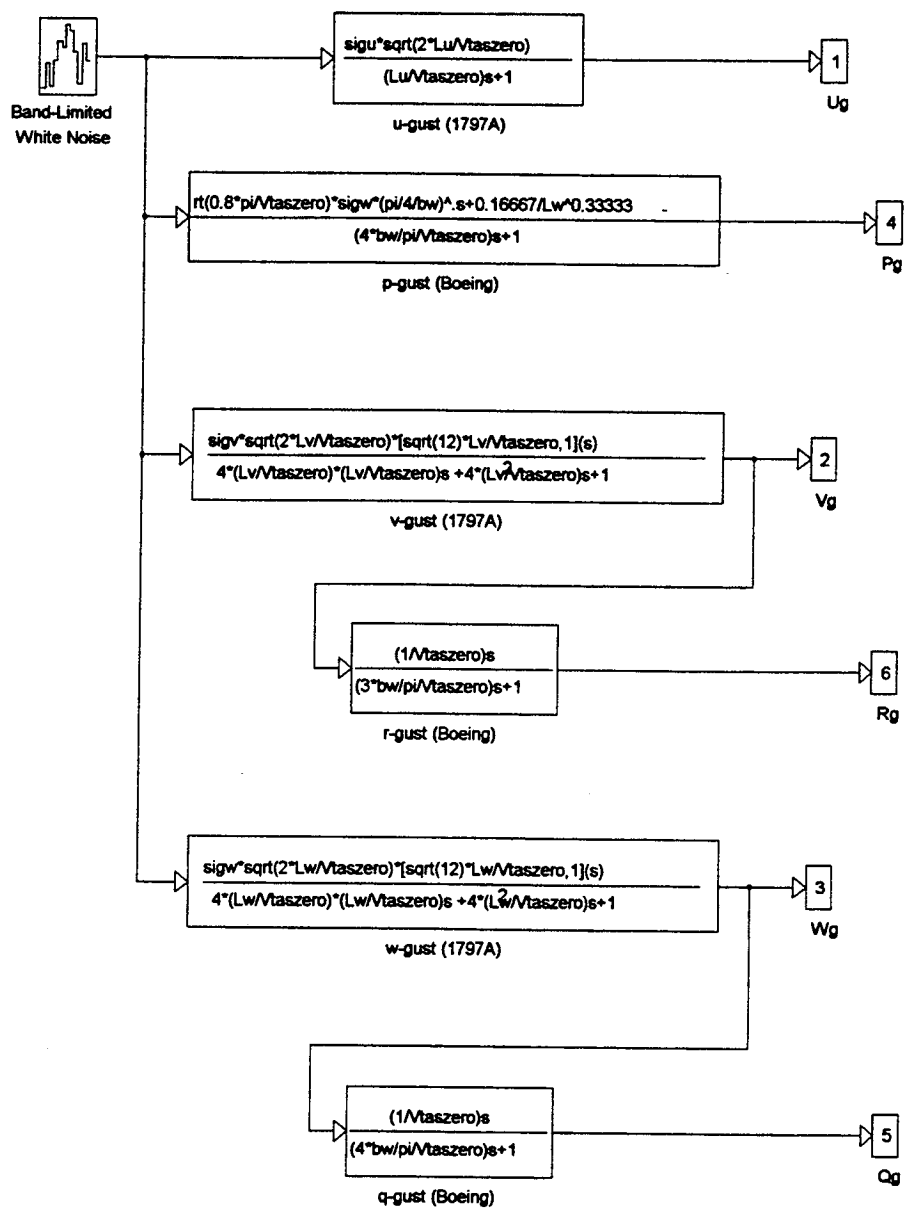
Define constants VwindN, VwindE, VwindD in the MATLAB workspace before invoking this model



Wind subsystem

## DRYDEN WIND GUST MODEL

Note: Initial airspeed is used  
in gust calculation.



Dryden wind gusts subsystem

<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1996		3. REPORT TYPE AND DATES COVERED Contractor Report
4. TITLE AND SUBTITLE Fuzzy Logic Decoupled Longitudinal Control For General Aviation Airplanes			5. FUNDING NUMBERS NCA1-113 WU 538-07-11-01	
6. AUTHOR(S) Noel Duerksen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Raytheon Aircraft Comany PO Box 85 Wichita, KS 67201-0085			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA CR-201639	
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Kenneth H. Goodrich				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category - 08			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>It has been hypothesized that a human pilot uses the same set of generic skills to control a wide variety of aircraft. If this is true, then it should be possible to construct an electronic controller which embodies this generic skill set such that it can successfully control difference airplanes without being matched to a specific airplane.</p> <p>In an attempt to create such a system, a fuzzy logic controller was devised to control throttle position and another to control elevator position. These two controllers were used to control flight path angle and airspeed for both a piston powered single engine airplane simulation and a business jet simulation. Overspeed protection and stall protection were incorporated in the form of expert systems supervisors.</p> <p>It was found that by using the artificial intelligence techniques of fuzzy logic and expert systems, a generic longitudinal controller could be successfully used on two general aviation aircraft types that have very difference characteristics. These controllers worked for both airplanes over their entire flight envelopes including configuration changes. The controllers for both airplanes were identical except for airplane specific limits (maximum allowable airspeed, throttle lever travel, etc.). The controllers also handled configuration changes without mode switching or knowledge of the current configuration.</p> <p>This research validated the fact that the same fuzzy logic based controller can control two very different general aviation airplanes. It also developed the basic controller architecture and specific control parameters required for such a general controller.</p>				
14. SUBJECT TERMS general aviation, flight controls, fuzzy logic, and decoupled control			15. NUMBER OF PAGES 105	
			16. PRICE CODE A06	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	